CLUBE Z80

Fevereiro/85

N.º 29

Coordenadores: ANA PAULA • ALEXANDRE SOUSA • J. MAGALHAES • AV.ª BOAVISTA, 832-2.º T. • TELEF. 65127 • 4100 PORTO

NESTE NÚMERO

No interior:

Folheto Mercado Z80

Edição: Clube Z80

Fotocomposição: Fotomecânica Mabreu/Porto

Impressão: Ramos dos Santos & C.ª, Lda./Porto

Tiragem: 500 exemplares, Fevereiro 1985

INTRODUÇÃO À LINGUAGEM MÁQUINA

ZX81/SPECTRUM

Autor: FERNANDO PRECES
SACAVÉM

(Cont. dos números anteriores)

SUPLEMENTO (SÉRIE MONITORES)

O ASSEMBLADOR

Este programa permite ao utilizador, partindo de instruções introduzidas pelo teclado ou de um programa previamente gravado em cassete, elaborar rotinas em código máquina, sem ter o trabalho de as codificar.
É normal, para não dizer indispensável, a construção dum diagrama bloco que represente todas as variantes principais dum programa que se pretende criar.
Depois começa-se por agrupar as mnemónicas que possam executar as funções que destinámos a cada rotina. E é neste ponto que o nosso assemblador se torna num colaborador precioso.
Ele vai transformar essas mnemónicas nos seus respectivos códigos, carregá-los na memória a partir do endereço indicado, desde que o utilizador se não esqueça de cumprir as principais normas que o programa estabelece para uma correcta recepção de dados.
Cada linha de dados, que é sempre referenciada por um endereço, possui 4 campos:

1 — Nomes (Labels ou etiquetas)
2 — Campo da mnemónica
3 — » de endereçamento
4 — » de comentários

Os campos dos nomes e dos comentários são opcionais.
O campo do endereçamento pode conter um endereço, um dado, ou estar vazio. O campo da mnemónica **nunca pode estar vazio**; ou tem a mnemónica ou tem uma directiva.
A separação entre os diferentes campos é feita por delimitadores.
O mais comum entre eles é o **espaço.**
O padrão deste assemblador para delimitadores é o seguinte:

:	— Após um NOME.
espaço	— No campo da mnemónica, a separar a acção do nome do registo, do endereço ou do dado.
;	— Antes dum comentário.

Quando um nome é colocado no respectivo campo, o assemblador associa-o ao endereço do primeiro código da mnemónica a escrever nessa linha. Sempre que pretender referir esse endereço, basta indicar o nome.
Os nomes devem ser curtos e indicarem muito claramente a acção que se pretende especificar.
O campo da mnemónica é sem dúvida o mais importante, visto ser o responsável pela formação do código máquina que o assemblador vai produzir.
Em substituição da mnemónica, este campo pode receber algumas directivas, tais como:

ORG (de origem)
EQU (de igualar)
DEF (de definir)

A directiva ORG permite localizar um programa, uma rotina, ou mesmo um quadro ou tabela de dados, em qualquer ponto da memória. Para que o assemblador associe essa directiva ao endereço pretendido é necessário que ela seja colocada como primeira instrução desse mesmo grupo.
Exemplo:

```
P1 :  ORG  50000
P2 :  ORG  55000
```

A directiva EQU associa o valor do seu argumento (um endereço) ao nome que a antecede e também deve ser colocada no início de cada programação.
Exemplo:

```
GRAF :  EQU  48000
```

A directiva DEF é formada por 2 comandos distintos:

1 — DEFB
2 — DEFW

DEFB Define um código entre (0 e 255)
DEFW Define um código entre (0 e 65535)

Esta directiva tem de ser utilizada sempre que seja necessário introduzir dados nos respectivos endereços.
Assim:

DEFB 16 — coloca o código 16 no endereço dessa linha
DEFW 23760 — coloca os dois bytes que formarão esse número, no endereço dessa linha e seguinte.

O campo de endereçamento consente a entrada de endereço ou dado, representados em decimal ou hexadecimal.

Exemplo: (endereço) 16384 ou 6400h
(dado) 156 ou 9Ch

O campo de comentários consente a introdução de qualquer tipo e comentário, desde que esteja antecedido de (;).

Exemplo: ; último quadro

O assemblador é formado por um programa em Basic, que começa na linha 1000. Uma vez carregado, ele vai pedir um valor para a RAMTOP, que se pode situar entre o mínimo indicado e o topo da memória, dependente, claro está, da quantidade de código máquina a utilizar.
Como o programa pode trabalhar com uma Printer (escrevente), quando formulada a pergunta, terá de responder (S) ou (N).
À seguinte, «PESQUISA AUTOMÁTICA?», também a resposta é (S) ou (N), em que (n) corresponde à entrada do assembler pelo teclado e (s) vai introduzir mais 1 questionário. «Digase o programa está gravado? (S ou N)». Se a resposta for (S) pedirá o nome do programa para o carregar.
A resposta (N) exige que o programa a assemblar tenha sido previamente escrito e esteja já à disposição do assemblador. Descubra você mesmo, caro leitor, a razão de ser destes 3 formatos de entrada, quando pretende codificar o seu assembler.

Em pesquisa auto, o programa quando encontra um erro, fará STOP, para que este possa ser emendado continuando a sua execução com GOTO 1020.

Se estiver a introduzir o assembler pelo teclado, um erro pode ser emendado, repetindo a linha já devidamente corrigida.

Como escrever um programa a assemblar no espaço compreendido entre as linhas 0 e 999:

```
  1 REM (org 40000
  5 REM SALT 1 : equ 45000
 10 REM SALT 2 : equ 46000
 15 REM CONT : defb 31
 20 REM Q1 : defw 45800
 25 REM Q2 : defw 45900
 30 REM Ld b cont
 35 REM Ld hl q1
 40 REM loop 1 : Ld a (hl)
 45 REM rst 16 ; rom écran
 50 REM inc hl
 55 REM dec b
 60 REM Ld a b
 65 REM Cp 0
 70 REM jr nz loop 1
 .....   .....
998 REM ret ; final
999 REM )
```

NOTA: Os programas podem ser todos escritos em minúsculas. Têm de abrir com «(» código 40 e fechar com «)» código 41, para que o pesquisador auto reconheça aonde começa a assemblagem e aonde acaba.

Em execução manual, entrada directa pelo teclado, não introduza os sinais de abertura ou fecho, bem como o número de linha e a instrução REM. Para terminar basta deixar a linha em branco e premir ENTER.

Para gravar o código máquina produzido pelo assemblador, responda ao questionário apresentado no final.

F I M

```
1000 REM   Assemblador para o ZX Spectrum
1001 REM  DIDACTIC SOFTWARE
1002 PRINT AT 10,0;" Indique um valor para a RAMTOP,''"        nao inferior a 34999": INPUT r$: CLEAR VAL r$
1005 LET j=VAL "1": IF PEEK VAL "23731">VAL "127" THEN CLS : PRINT AT 10,7;"RAMTOP VALIDA.": PAUSE 200: GO TO 1010
1008 CLS : PRINT AT 10,6;"INVALIDA RAMTOP.": PAUSE 100: CLS : GO TO 1002
1010 LET ns=64: LET nu=64
1015 CLS : PRINT "*** ZX Spectrum Assembler ***": PRINT '"Avaliacao RAM: ";(PEEK 23733-63)/4;"K"''"Valor RAMTOP : ";: LET h=FN i(23730): GO SUB 9100: PRINT h
1020 DEF FN a(X$)=(X$>="A" AND x$<="Z")
1030 DEF FN n(X$)=(X$>="0" AND x$<="9")
1035 DEF FN s$(x)=a$(x(x,j)+j TO x(x,2))
1040 DEF FN h(x)=INT (x/256)
1050 DEF FN l(x)=x-256*FN h(x)
1060 DEF FN j()=(m$="JP" OR m$="CALL" OR m$="RET" OR m$="JR")
1070 DEF FN v(x$)=CODE x$(7)+256*CODE x$(8)
1080 DEF FN i(x)=PEEK x+256*PEEK (x+j)
1090 DIM s$(ns+nu+j+j,VAL "8"): DIM l$(VAL "6"): DIM t$(VAL "2")
1095 LET s$(j)=CHR$ NOT j
1100 LET u=NOT j: LET er=u: LET ed=VAL "237"
1110 LET m$="Com copia?": GO SUB 9000: LET hc=j+j+(CHR$ CODE a$="s")
1120 LET m$="Pesquisa automatica?": GO SUB 9000: LET auto=(CHR$ CODE a$="s"): IF auto THEN LET m$="Os codigos estao gravados?": GO SUB 9000: IF CHR$ CODE a$="s" THEN PRINT ''" As linhas desse programa terao de ser numeradas entre 0 e 999.": INPUT "Qual o Nome? : "; LINE a$: CLS : PRINT AT 8,0;"LIGUE O GRAVADOR PRIMA UMA TECLA": PAUSE 4e4: CLS : MERGE a$
1125 IF auto THEN GO SUB 7800
1130 LET add=FN i(VAL "23730")+VAL "1"
1140 CLS : PRINT #hc;"Loc    Cod.      Mnemonicas     "''
1200 LET e=NOT j: LET vdef=j: LET dl=e: LET dd=e: LET ec=e: LET l=e: LET w=e
1210 LET h=add: GO SUB 9100: LET m$=h$: GO SUB 7700: IF end THEN GO TO VAL "5000"
1220 LET m$=""
1230 DIM t(2): DIM x(4,2): LET t=NOT j: LET p=j: LET t$="00"
1240 FOR x=j TO LEN a$
1245 IF a$(x)=";" THEN GO TO 1300
1250 IF t<>(a$(x)<>" " AND a$(x)<>",") THEN LET x(p,t+j)=x-j: LET p=p+t: LET t=NOT t: IF p>4 THEN GO TO 1310
1260 NEXT x
1300 IF x(p,j) OR p=j THEN LET x(p,j+j)=x-j
1310 LET lx=(FN s$(j)(LEN FN s$(j))=":")
1320 LET m$=FN s$(j+lx)
1400 LET jump=FN j()
1410 FOR t=j TO j+j: LET x$=FN s$(j+t+lx): GO SUB 7400: NEXT t
1500 LET n$=t$+m$
1510 LET l1=(LEN FN s$(lx+2)<>0): LET l2=(LEN FN s$(lx+3)<>0): IF l1 THEN GO TO 1600
1520 IF m$="" THEN GO TO 1900
1530 FOR l=j TO j+j: LET i=VAL "2010+l*10": GO SUB 8300: IF NOT x THEN NEXT l
1550 GO TO 1630
1600 LET i=VAL "2060+l2*20": LET l=NOT j
1610 GO SUB 8300: IF NOT x+l2 THEN LET i=2070: GO SUB 8300
1615 IF x THEN GO TO 1650
1620 RESTORE VAL "2000+40*(l2=0)": GO SUB 8400: IF x THEN GO TO 1800
1630 IF NOT x THEN LET ec=VAL "9"
1640 GO TO 1800
1650 LET n=x-j: RESTORE i+j: LET n$=t$: GO SUB 8400: IF NOT x THEN LET ec=VAL "9"
1800 IF ec THEN GO TO 1960
1820 IF m$="ORG" THEN LET add=di: LET h=di: GO SUB 9100
1830 IF e THEN POKE add,dd: IF e=j+j THEN POKE add+j+j,dis
1840 FOR x=0 TO l-j: READ x$: POKE add+x+(e<>0)+x*(e=2),VAL x$: NEXT x
1900 IF NOT vdef THEN GO SUB 7200
1910 IF NOT lx OR ec THEN GO TO 1960
1920 LET l$=FN s$(lx)( TO LEN FN s$(lx)-j): GO SUB 7600: IF v THEN LET ec=2: GO TO 1960
1925 IF s=ns+j THEN LET ec=VAL "
```

```
99": GO TO 1960
1930 LET v=add: IF m$="EQU" THEN
 LET v=di
1940 LET s$(s)=l$: LET s$(s,7 TO
 )=CHR$ FN l(v)+CHR$ FN h(v)
1950 LET s$(j)=CHR$ (s-j)
1960 GO SUB 7300: GO SUB 1995: I
F ec THEN LET er=er+j: GO TO VAL
 "1200"
1970 LET add=add+l+e
1990 GO TO 1200
1995 IF ec AND auto THEN PRINT '
'"Emende o erro ": STOP
1998 RETURN
2000 DATA "11LD","1","1","64+T(1
)*8+T(2)"
2001 DATA "13LD","vdef","2","t(1
)*8+6","dl"
2002 DATA "14LD","t(1)=7","3","5
8","dl","dh"
2003 DATA "15LD","(t(1)=7)*(t(2)
<2)","1","t(2)*16+10"
2004 DATA "23LD","t(1)<>6","3","
t(1)*16+1","dl","dh"
2005 DATA "24LD","t(1)=2","3","4
2","dl","dh"
2006 DATA "24LD","t(1)<4","4","e
d","75+t(1)*16","dl","dh"
2007 DATA "41LD","t(2)=7","3","5
0","dl","dh"
2008 DATA "42LD","t(2)=2","3","3
4","dl","dh"
2009 DATA "42LD","t(2)<4","4","e
d","67+t(2)*16","dl","dh"
2010 DATA "51LD","(t(2)=7)*(t(1)
<2)","1","t(1)*16+2"
2011 DATA "99","","2100"
2020 DATA "RET","201","NOP","0",
"RLCA","7","RRCA","15","RLA","23
","RRA","31","DAA","39","CPL","4
7","SCF","55","CCF","63","HALT",
"118","EXX","217","DI","243","EI
","251","99",""
2030 DATA "NEG","ED","68","RETN"
,"ED","69","RETI","ED","77","RRD
","ED","103","RLD","ED","121","L
DI","ED","160","LDIR","ED","176"
,"LDD","ED","168","LDDR","ED","1
84","CPD","ED","169","CPDR","ED"
,"185","CPI","ED","161","CPIR","
ED","177","99","",""
2040 DATA "30JR","1","2","24","v
def*FN l(di-add-2)"
2041 DATA "10INC","1","1","4+T(1
)*8"
2042 DATA "20INC","T(1)<>4","1",
"3+T(1)*16"
2043 DATA "10DEC","1","1","5+T(1
)*8"
2044 DATA "20DEC","T(1)<>4","1",
"11+T(1)*16"
2045 DATA "30DJNZ","1","2","16",
"vdef*FN l(di-add-2)"
2046 DATA "60RET","1","1","192+T
(1)*8"
2047 DATA "20POP","T(1)<>3","1",
"193+T(1)*16-16*(T(1)=4)"
2048 DATA "20PUSH","T(1)<>3","1"
,"197+T(1)*16-16*(T(1)=4)"
2049 DATA "30RST","vdef AND dl<=
56","1","199+INT (dl/8)*8"
2050 DATA "10JP","t(1)=6","1","2
33"
2051 DATA "30IM","vdef*(dl<3)*(d
l>=0)","2","ed","70+16*(dl=1)+24
*(dl=2)"
2052 DATA "30JP","1","3","195","
dl","dh"
2053 DATA "30CALL","1","3","205"
,"dl","dh"
2054 DATA "30DEFB","vdef","1","d
l"
2055 DATA "30DEFW","vdef","2","d
l","dh"
2056 DATA "30ORG","vdef","0"
2057 DATA "30EQU","vdef AND lx",
"0"
2059 DATA "99","","0"
2060 DATA "","","SUB","","AND","
XOR","OR","CP","99"
2061 DATA "10","1","1","128+n*8+
T(1)"
2062 DATA "30","vdef","2","198+n
*8","dl"
2063 DATA "99","","0"
2070 DATA "RLC","RRC","RL","RR",
"SLA","SRA","","SRL","99"
2071 DATA "10","1","2","203","n*
8+t(1)"
2072 DATA "99","","0"
2080 DATA "ADD","ADC","","SBC","
99"
2081 DATA "11","t(1)=7","1","128
+n*8+t(2)"
2082 DATA "13","t(1)=7","2","198
+n*8","dl"
2083 DATA "22","(t(1)=2)*(t(2)<>
4)*(n=0)","1","9+t(2)*16"
2084 DATA "22","(t(1)=2)*(t(2)<>
4)*(n<>0)","2","ed","78-n*4+t(2)
*16"
2085 DATA "99","","0"
2100 DATA "63JP","1","3","194+t(
1)*8","dl","dh"
2101 DATA "63JR","t(1)<4","2","3
2+t(1)*8","vdef*FN l(di-add-2)"
2102 DATA "22EX","(t(1)=4)*(t(2)
=4)","1","8"
2103 DATA "22EX","(t(1)=1)*(t(2)
=2)","1","235"
2104 DATA "52EX","(t(1)=3)*(t(2)
=2)","1","227"
2112 DATA "31BIT","vdef AND dl<8
","2","203","64+dl*8+t(2)"
2113 DATA "31RES","vdef AND dl<8
","2","203","128+dl*8+t(2)"
2114 DATA "31SET","vdef AND dl<8
","2","203","192+dl*8+t(2)"
2117 DATA "63CALL","1","3","196+
t(1)*8","dl","dh"
2120 DATA "14IN","vdef AND t(1)=
7","2","219","dl"
2121 DATA "41OUT","vdef AND t(2)
=7","2","211","dl"
2122 DATA "17IN","t(1)<>6","2","
ed","64+t(1)*8"
2123 DATA "71OUT","t(2)<>6","2",
"ed","65+t(2)*8"
2200 DATA "99","","0"
5000 IF CODE s$(j) THEN PRINT #h
c''"Simbolos:"': FOR y=VAL "2" T
O VAL "CODE s$(1)+1": LET h=FN v
(s$(y)): GO SUB 9100: PRINT #hc;
TAB VAL "((y-2)*16+1)";s$(y, TO
6);" ";h$;: NEXT y
5010 LET n=NOT j
5020 FOR t=ns+3 TO ns+2+u: LET a
dd=FN v(s$(t)): LET c=PEEK add:
LET l$=s$(t): GO SUB 7600: IF v
THEN GO TO 5050
5030 IF NOT n THEN PRINT #hc''"E
RRO :"
5040 LET n=n+j: LET h=add: GO SU
B 9100: PRINT #hc;" ";l$;" ";h$:
 GO TO 5100
5050 RESTORE 9920: LET di=FN i(a
dd+j): LET dl=FN l(di): LET di=V
AL "di-65536*(di>32767)"
5055 LET dl=VAL "dl-256*(dl>127)
"
5060 READ c$,a$,x$: IF NOT VAL c
$ THEN FOR x=j TO VAL x$: READ x
$: NEXT x: GO TO 5060
5070 FOR x=j TO VAL a$: READ y$:
 POKE FN v(s$(t))+VAL x$+x-j,VAL
 y$: NEXT x
5100 NEXT t
5200 IF er+n THEN PRINT #hc'er+n
;" Erro(s)": GO TO 5300
5210 GO TO 9990
5300 INPUT "Ha erros a corrigir
? (s ou n) ";l$
5310 IF l$="s" THEN INPUT "Ender
eco ? ";w;"Codigo ? ";u: POKE w,
u: GO TO 5300
5320 INPUT "Quer gravar o C/M ?
(s ou n) ";l$
5330 IF l$="s" THEN GO TO 9990
```

```
5340 RUN
7200 REM indefinido
7210 LET u=u+j: IF u>nu THEN LET
 ec=VAL "98": RETURN
7220 LET s$(ns+u+j+j)=s$(ns+j+j,
 TO 6)+CHR$ FN l(add)+CHR$ FN h(
add)
7230 RETURN
7300 REM Escrita
7310 IF ec THEN PRINT #hc;h$;" "
; INVERSE j;"ERRO ";ec; INVERSE
NOT j;TAB 14;a$: RETURN
7320 IF lx THEN PRINT #hc;h$;TAB
 VAL "13";FN s$(j)
7330 IF m$="" THEN RETURN
7340 PRINT #hc;h$;"■" AND (NOT v
def);TAB 6;: FOR y=0 TO l+e-j: L
ET h=PEEK (add+y): GO SUB 9120:
PRINT #hc;h$;: NEXT y
7350 PRINT #hc;TAB 14;m$;TAB 19;
FN s$(j+j+lx);"," AND l2;FN s$(3
+lx)
7360 RETURN
7400 REM check type
7405 LET x=LEN x$: IF NOT x THEN
 RETURN
7410 LET ix=(x$(j)="("): IF ix T
HEN LET x$=x$(2 TO LEN x$-j)
7415 GO SUB 7500
7420 RESTORE 9900+jump: READ n:
FOR x=0 TO n-j: READ c$: IF x$=c
$ THEN GO TO 7450
7425 NEXT x
7430 LET t$(t)=("3" AND (NOT ix)
)+("4" AND ix)
7440 GO SUB 8500: LET di=a+w: LE
T dh=FN h(di): LET dl=FN l(di):
RETURN
7450 LET t(t)=x: IF jump AND (NO
T ix) THEN LET t$(t)="6": RETURN

7460 LET t$(t)="11111111222222220
7000000551501l"(ix*15+x+j)
7470 IF x>=13 THEN LET e=j+ix-ju
mp: LET dd=VAL "dd+(dd=0)*(221+3
2*(x=14))": LET dis=w: LET x=10
7475 IF x>7 THEN LET t(t)=x-8
7480 IF ix AND t$(t)="1" THEN LE
T t(t)=6
7490 RETURN
7500 FOR x=j+j TO LEN x$: IF x$(
x)="+" OR x$(x)="-" THEN GO TO 7
550
7510 NEXT x
7520 LET w=NOT j: RETURN
7550 LET z$=x$( TO x-j): LET x$=
x$(x TO ): GO SUB 8900: LET x$=z
$
7560 LET w=x
7570 RETURN
7600 REM find symbol
7610 LET v=j
7620 FOR s=2 TO CODE s$(j)+j: IF
 l$=s$(s, TO 6) THEN LET a=FN v(
s$(s)): RETURN
7630 NEXT s: LET v=0
7640 RETURN
7700 IF NOT auto THEN GO TO 9000
7710 LET end=0: IF FN i(auto-1)=
10730 THEN LET end=1: RETURN
7720 LET a$=""
7730 IF PEEK auto=13 THEN GO TO
7760
7735 IF PEEK auto=33 THEN LET au
to=auto+1: GO TO 7770
7740 LET a$=a$+CHR$ PEEK auto
7750 LET auto=auto+1: GO TO 7730
7760 LET auto=auto+6
7770 IF NOT LEN a$ THEN GO TO 77
10
7780 GO TO 9020
7800 REM Arranque em auto
7810 PRINT AT 10,8;"Em pesquisa.
..": PAUSE 50
7820 LET x=FN i(23635): LET y=FN
 i(23627)
7830 IF x>=y THEN LET auto=0: RE
TURN
7840 IF PEEK (x+4)=234 THEN GO T
O 7860
7850 LET x=x+4+FN i(x+2): GO TO
7830
7860 IF PEEK (x+5)<>40 THEN GO T
O 7850
7870 LET auto=x+6
7880 RETURN
8300 REM Find op (1)
8310 RESTORE i: LET x=0
8320 READ x$: IF x$="99" THEN LE
T x=0: RETURN
8330 LET x=x+j: IF m$=x$ THEN RE
TURN
8340 FOR y=j TO l: READ x$: NEXT
y
8350 GO TO 8320
8400 REM Find op (2)
8410 READ x$,c$,y$: LET z=VAL y$
: IF x$="99" THEN GO TO 8450
8420 IF x$=n$ THEN LET x=VAL c$:
 IF x THEN LET l=z: RETURN
8430 FOR y=j TO z: READ x$: NEXT
y
8440 GO TO 8410
8450 LET x=0: IF NOT z THEN RETU
RN
8460 RESTORE z: GO TO 8410
8500 REM Convert type
8505 LET a=0
8510 GO SUB 8900: IF v THEN LET
a=x: LET ec=VAL "6*(x>65535 OR x
<-32768)": RETURN
8520 IF FN n(x$(j)) AND x$(LEN x
$)="H" THEN FOR x=j TO LEN x$-j:
 LET a=VAL "a*16+CODE x$(x)-48-7
*(x$(x)>""9"")": NEXT x: RETURN
8530 LET l$=x$: GO SUB 7600: IF
v THEN RETURN
8560 LET ec=VAL "6*(FN a(x$(1))=
0)": IF ec THEN RETURN
8570 LET s$(ns+2)=x$: LET vdef=0
8580 RETURN
8900 REM vet numeric
8910 LET z=j+VAL "((x$(1)=""+"")
 OR (x$(1)=""-"")) AND LEN x$>1"
8920 FOR n=z TO LEN x$: IF FN n(
x$(n)) THEN NEXT n
8930 LET v=(n>LEN x$): IF v THEN
 LET x=VAL x$( TO n-j)
8940 RETURN
9000 REM kybrd
9010 INPUT (m$+" "); LINE a$: IF
 a$="s" THEN RETURN
9015 LET end=NOT LEN a$
9020 FOR x=j TO LEN a$: LET a$(x
)=CHR$ (CODE a$(x)-32*(a$(x)>="a
")): NEXT x
9030 RETURN
9100 REM h$=hex$(h)
9110 LET h$="     ": GO TO 9150
9120 LET h$="  "
9130 LET h1=h: FOR x=LEN h$ TO j
 STEP -j: LET x1=h1-INT (h1/16)*
16: LET h$(x)=CHR$ (x1+CODE "0"+
7*(x1>9)): LET h1=INT (h1/16): N
EXT x
9140 RETURN
9150 LET h1=h: FOR x=LEN h$ TO j
 STEP -j: LET x1=h1-INT (h1/10)*
10: LET h$(x)=CHR$ (x1+CODE "0"+
7*(x1>9)): LET h1=INT (h1/10): N
EXT x
9160 RETURN
9900 DATA 15,"B","C","D","E","H"
,"L","M","A","BC","DE","HL","SP"
,"AF","IX","IY"
9901 DATA 15,"NZ","Z","NC","C","
PO","PE","P","M","","","HL","","
","IX","IY"
9920 DATA "(c-INT (c/8)*8)+INT (
c/64)=0","1","1","dl+FN l(a-add-
2)"
9921 DATA "c=ed OR c=221 OR c=25
3","2","2","FN l(a+di)","FN h(a+
di)"
9922 DATA "1","2","1","FN l(a+di
)","FN h(a+di)"
9990 INPUT "Nome do Programa ? "
;w$: INPUT "codigo inicial ? ";y
$: INPUT "Extensao ?";x$
9995 SAVE w$CODE VAL y$,VAL x$
9998 STOP
9999 SAVE "Assembl 5" LINE 1000
```

PROGRAMAÇÃO

Linguagem FORTH

De quando em quando somos solicitados a abordar uma das novas linguagens de que os nossos amigos começam a ouvir falar com insistência.
Hoje cabe a vez à linguagem FORTH (pronunciar FORSSS) pouco conhecida ainda, quer dos amadores (que somos todos nós) quer dos profissionais; é preciso confessar que se trata de uma linguagem muito particular e em todo o caso bastante diferente das outras.
Devo dizer-vos, à partida, que é mais fácil para quem não conhece nenhuma linguagem, aprender rapidamente a usar o FORTH. Quem já conhece outras linguagens, tem sempre tendência em procurar uma semelhança ou uma analogia com outra linguagem e esbarra com uma impossiblidade... não existe paralelismo entre os conceitos do FORTH e, por exemplo, o BASIC!
Para os amigos do Clube Z80 não é difícil a iniciação nesta linguagem pois ela está disponível para o ZX SPECTRUM ou para o APPLE, TRS80, BBC, etc.
FORTH nasceu nos anos 70, nos Estados Unidos e o seu criador é um programador chamado CHARLES MOORE. Este homem, dotado de grande engenhosidade, tinha necessidade e possuir um dispositivo que lhe testasse rapidamente os programas que projectava; como nenhuma das linguagens que ele dominava lhe convinha, Moore concebeu e escreveu o primeiro compilador FORTH.
O nome da linguagem provém da constatação de que era de tal forma diversa das que existiam, que foi classificada como da quarta (fourth em inglês) geração.
É interessante notar que durante vários anos, Charles Moore foi o único utilizador desta linguagem, num computador IBM 1130 e como esta máquina apenas podia comportar 5 caracteres para um identificador, em vez de FOURTh ficou FORTH.
Observando os conceitos e a forma da linguagem FORTH, deve constatar-se que em certos aspectos se parece com a linguagem máquina (daí a sua rapidez!), mas diverge no sentido de que não possui as dificuldades de escritura nem as limitações da linguagem máquina. Ela consegue ser linguagem programação interpretada, compilada, monitor, editor mesmo assemblador.
O Compilador FORTH é extremamente compacto comparado com outras linguagens, pois vai de 2 Kbytes até 8 Kbytes (neste caso será já de luxo!).
Este pormenor implica que também os programas já compilados sejam de tal modo comprimidos, que uma máquina Forth com 2 K de memória RAM é equivalente a uma máquina BASIC com 16 K.
Forth é uma linguagem recursiva, particularmente adaptada às aplicações tipo multiprogramação ou tempo real (como a robótica, por exemplo).
É uma linguagem estruturada; não é possível encontrar o GOTO em Forth mas, em contrapartida, podemos ter ciclos imbricados uns nos outros até ao infinito.
Utiliza uma PILHA ou STACK para todas as suas operações e instruções; pilha esta que permite ignorar totalmente o que se passa na memória e onde isso tem lugar, pois que apenas a PILHA tem importância.
FORTH é uma linguagem interactiva que vos permite testar imediatamente cada peça de programa dado que a maioria das «instruções» FORTH pode ser executada imediatamente.
Podemos aceder à linguagem máquina sem qualquer dificuldade ler ou escrever na memória ou num registo de circuito (trata-se sempre com operações elementares).
A inclusão de programas em linguagem máquina no interior de programas FORTH é imediata e passa desapercebida de tal modo ela é natural e simples; além disso FORTH, dispõe de um assemblador integrado para facilitar este trabalho.
Toda a medalha tem um reverso, FORTH não escapa a esta regra e possui as suas limitações.
Inaptidão para o cálculo científico e dificuldades em manipular grandes palavras... 65535 é já um número muito grande para FORTH. Não esquecer que as linguagens são sempre concebidas para... isto... ou para aquilo..., portanto continuem calcular os valores trigonométricos e o orçamento familiar em BASIC.
Certos programadores dizem com uma certa piada, que o FORTH foi criado para ser escrito e não para ser lido, dado que existe uma certa dificuldade em ler os programas em FORTH.
Quem resistiu até agora e continua curioso(a) vai ver que existem coisas interessantes; por exemplo esqueça as instruções e as funções! O Forth apenas possui: PALAVRAS.... uma palavra Forth pode ser DROP ou ROT, mas também pode ser HOJE ou ONTEM ou ainda %?/3a I I b que tudo são palavras correctas.
De facto, para ser preciso, Forth dispõe à partida (quando liga a máquina) de um certo número de palavras contidas no seu dicionário mas isto não é limitativo e pode ser levado até ao infinito.... (desde que a memória também o seja)!
Todas as palavras que tu criares são acrescentadas ao DICIONÁRIO original, ou seja, falando em termos do nosso universo de Basic, podes criar as tuas próprias «instruções» que farão em seguida parte do jogo de «instruções» para posterior uso, não importa por quem ou quando.

Estas palavras que tu podes criar são de facto programas FORTH; com efeito, escrever um programa Forth consiste em usar palavras do Dicionário, realizando uma função bem definida e a «assemblar» essas funções de modo a formar novas palavras até obter a função desejada. Por outros termos, cosntruímos o nosso programa bloco a bloco e cada bloco irá realizar uma função bem definida, por ínfima que seja. É divertido pensar que o meu programa, uma vez terminado, será uma palavra que fará parte do dicionário e que pode ser usado por qualquer outro como uma vulgar «instrução».
Dissemos atrás que FORTH usa uma PILHA, mas antes de ver como e porquê, examinemos de perto este conceito de PILHA.
A pilha Forth está definida na memória RAM como devem ter percebido, mas a sua colocação (ou a definição dos seus limites) não nos interessam; o que é importante é saber como ter acesso à pilha em qualquer momento.
Imaginemos uma pilha de folhas de papel à qual podemos juntar outras folhas ou retirá-las, mas sempre na mesma ordem. Por exemplo, se eu tenho uma série de encomendas, devo satisfazer os pedidos pela ordem de chegada... neste

tipo de pilha, o último pedido a chegar será o primeiro a partir (satisfeito); trata-se de uma estrutura LIFO (last in last out). Por aqui se pode ver que a parte fundamental da pilha será sempre a camada superior (isto em sentido figurado).
Esta Pilha vai ser usada por FORTH para guardar dados, constantes, variáveis dos ciclos, resultados; em resumo: TUDO!
A PRIMEIRA PALAVRA... será PONTO (.) ou seja o DOT dos ingleses — o ponto serve para obter no écran o valor que está na parte superior da pilha e que será destruído mal o retiramos da pilha.
Use o compilador fig-FORTH que está disponível para o Spectrum, reproduza do gravador com LOAD " ", e mal ele fique instalado, experimente:

4 enter 3 enter 2 enter 1 enter

A tua máquina responde sempre ok e vai colocando cada um destes valores no interior da pilha, pela ordem de entrada. Escreve agora o ponto e enter.
Observa que terás: 1 ok novo ponto e enter e terás 2 ok, etc., ou seja, vais retirando os valores que havias introduzido previamente na pilha.
Quando a pilha está vazia e tentamos obter valores usando: . enter teremos a mensagem: 46.? MSG I I 1 que significa PILHA VAZIA.
Regras de Sintaxe > são simples, mas todavia é preciso conhecê-las. Podemos escrever dados e palavras Forth seguidas umas às outras na mesma linha, desde que separadas por um espaço.
A interpretação das linhas FORTH apenas começa após o <enter> que é universalmente conhecido como CR ou Carriage Return ou chr$(13).
As palavras Forth que posso definir devem comportar no máximo 31 caracteres que podem ter qualquer caracter exceptuando o espaço, o caracter 13 (enter), o código 127 e o nulo (código 0).
As regras de sintaxe limitam-se a isto!
Observamos anteriormente que a palavra ponto (.) tem o significado do nosso habitual PRINT mas destrói o conteúdo da pilha. Entretanto, cómo pode ser útil limpar o conteúdo da pilha sem o projectar no écran podemos usar DROP seguido de enter e o valor que está no TOP da pilha desaparece, sem sair no écran.
O compilador que o Clube distribui é o fig-FORTH Abersoft 1.1, e embora as folhas descritivas sejam simplórias, existe um grupo de cópias do livro STARTING WITH FORTH.
Neste momento Janeiro/85, os dados sobre utilizadores FORTH, ão os seguintes:

EDUCAÇÃO	2,4 %
PROGRAMAÇÃO COMERCIAL	6,3 %
DIVERSOS	10 %
AQUISIÇÃO DE DADOS/INSTRUMENTOS	31,3 %
SISTEMAS DE CONTROLO/ROBÓTICA	50 %

Um dos objectivos do Clube Z80 foi sempre o da tentativa de criar núcleos de interesse à volta das coisas novas da Informática.
O objectivo tem falhado, porque nunca encontramos um retorno. Se tu te interessas por FORTH, diz que estás presente e manda a tua colaboração. Não esperes por obra monumental, porque todos somos amadores, mas as coisas simples, por vezes, fazem despertar outros temerosos do «ridículo» e avançar um pouco o mundo. Se não fosse a ingenuidade de quem tem espírito criativo, ainda hoje se escrevia apenas com um pedregulho granítico!

Um abraço do Clube Z80

NÚMEROS — COMO USAR OS NÚMEROS EM BASIC

Para manejar com eficiência os seus programas, deve conhecer algo de substancial em relação aos números, e a forma como são tratados em Basic.
Existem dois aspectos a considerar:

1 — como o programador os vê
2 — como o computador vê os números

O modo como vê os números depende do formato que usa para a entrada dos dados numéricos, usando os comandos INPUT, LET e READ.
O mesmo tipo de formato será usado pelo computador para apresentar os resultados usando PRINT e LPRINT. Existem três tipos de formato de interesse imediato.

a) INTEIROS (números tais como -1, 0, 25 etc.)

b) NÚMEROS REAIS (aqueles que possuem parte fraccionária, tais como 10.625, -1.33,3.142 and 1/2)

c) E formato científico (por exemplo 0.32E10 ou -1.683E-12)

O Spectrum também permite que os números entrem sob a forma BINARIA usando o comando POKE, no entanto dado que esse domínio está ligado ao tema dos Gráficos Definidos pelo Utilizador, não será aqui abordado.
Para entrada e saída de dados numérico, os valores devem ser atribuídos às respectivas variáveis. As regras de atribuição de valores são simples e conhecidas.

1) Os nomes de variáveis numéricas simples (contendo um dado cada), podem ser formados por combinações de letras e números, desde que o primeiro carácter seja uma letra.

2) Uma variável numérica tipo ARRAY ou MATRIZ ou QUADRO deve possuir um nome formado por apenas uma letra, mas pode usar a mesma letra que tenha sido atribuida a uma variável simples.

```
PROGRAMA 1

 300 REM CONVERSAO PONTO FLUTUAN
TE
 310 INPUT X
 315 LET N=0
 320 LET S=SGN X
 330 LET X=ABS X
 340 IF X>=2↑32 THEN GO TO 400
 350 LET X=2*X
 360 LET N=N+1
 370 IF X<2↑31 THEN GO TO 350
 380 GO TO 500
 400 LET X=X/2
 410 LET N=N-1
 420 IF X>=2↑32 THEN GO TO 400
 500 LET A=INT (X/256↑3)
 502 PRINT 2↑32: PRINT X
```

```
 506 LET X=X-256↑3*A
 508 LET B=INT (X/256↑2)
 510 LET X=X-256↑2*B
 520 LET C=INT (X/256)
 530 LET D=X-256*C
 540 PRINT "EXPOENTE=";160-N
 550 PRINT "MANTISSA=";A-128*(S=
1);"  ";B;"  ";C;"  ";D
 560 GO TO 300
```

REPRESENTAÇÃO DOS NÚMEROS

O microprocessador Z80 usa um padrão de 8 bits por cada BYTE (palavra processada). Entretanto observe que todas as representações de dados quer sejam números quer sejam caracteres, são guardados sob a forma de Bytes.
Existem três processos que o computador usa para manipular os números e apresentá-los no écran:

1) INTEIROS
2) PONTO FLUTUANTE (vírgula flutuante . . .)
3) BCD (Binary Coded Decimal)

A aritmética tipo BCD é agora obsoleta, de tal modo os micros são agora concebidos, no entanto o programador em Assembler sabe que o Z80 suporta este tipo de representação.
Os leitore mais atentos a este tipo de pormenores, devem interrogar-se sobre o porquê de distinguir entre inteiros e ponto flutuante.

```
PROGRAMA 2

  10 REM DEMONSTRACAO INTEIRO/PO
NTO FLUTUANTE
  20 INPUT N
  30 LET T$="F"
  40 LET V=PEEK 23627+256*PEEK 2
3628
  45 IF PEEK (V+1)=0 THEN LET T$
="I"
  50 PRINT "NUMERO";N;" TIPO ";
  60 IF T$="I" THEN PRINT "INTEI
RO""5 BYTES:-"
  70 IF T$="F" THEN PRINT "PONTO
 FLUTUANTE"'" EXP    MANTISSA"
  80 FOR X=1 TO 5
  90 PRINT PEEK (V+X);"    ";
 100 NEXT X
 110 GO TO 20
```

DADOS TIPO INTEIRO

Um número inteiro será interpretado pelo BASIC do Spectrum como um valor entre -65535 e +65535.
Serão necessários 5 bytes de memória para guardar esse número inteiro, da seguinte forma:

Byte 1 = 0 (não usado)
Byte 2 = 0 (no caso de um inteiro positivo)
= 255 (para o caso de um negativo)
Byte 3 = Byte menos significativo (low)
Byte 4 = Byte mais significativo (high)
Byte 5 = 0 (não usado)

Os bytes 3 e 4 guardam o valor inteiro. Em termos práticos, isto significa que o número N será retido na memória da seguinte forma:

Para um inteiro positivo: Byte 3 = N – 256 * INT (N/256)
Byte 4 = INT (N/256)

Para um inteiro negativo:

Byte 3 = 65536 – N – 256 * INT (65536 – N)/256)
Byte 4 = INT ((65536 – N)/256)

Como vimos, dois destes bytes não são usados neste tipo de representação o que permite ao computador modificar o conteúdo em vírgula (ponto) flutuante em qualquer instante.

DADOS TIPO PONTO FLUTUANTE

Qualquer inteiro fora da gama especificada para o Spectrum, seja real ou tipo E (mais de 8 dígitos) vai ser guardado como Ponto Flutuante.
O sistema parte o número em dois, Expoente e Mantissa, com o ponto (vírgula) decimal «flutuando» de acordo com a máxima significância. Sob o ponto de vista do sistema decimal, o processo é facilmente entendido.
Por exemplo:
$0.005679 = 0.5679 * 10^{-2}$
$- 0.5679 * E^{-2}$
O que significa: Mantissa = 0.5679
Expoente = – 2
Os números binários tipo ponto flutuante são de difícil compreensão, no entanto o princípio é o mesmo.
O processo é demonstrado com certa «limpeza» no programa 1. Do mesmo modo que o anterior, temos 5 bytes que são designados para guardar cada número tipo ponto flutuante. O programa calcula o Expoente (byte 1) e a Mantissa (byte 2) originando o número X.
Com efeito, nós colocamos um ponto decimal binário, imediatamente à esquerda do primeiro dígito binário significativo (caso decimal).
Em binário, 4 bytes são necessários para obter a precisão máxima, e então nós expandimos ou contraímos a Mantissa (isto em sentido figurado), simultaneamente, de modo a alterar o expoente (ou ajustar), multiplicando ou dividindo repetidamente por 2.

SETUP linhas 300 a 340

INPUT X (número real)
N = O (número de multiplicações/divisões por 2)
S = sinal de X
X = valor absoluto de X
IF X < = 2 ^ 32 (overflow dos 32 bits . . .
ultrapassou a capacidade)
THEN saltar para DIVIDE
Continue com MULTIPLIQUE

MULTIPLIQUE linhas 350 a 370

X = 2 * X (duplicar o número)
N = N + 1 incrementar
IF X < 2 ^ 31 (primeiro de quatro bytes
não está ainda activado
THEN saltar para MULTIPLIQUE
saltar para BYTE

DIVIDE linhas 400 a 430

X = x/2 dividir por 2
N = N – 1 decrementar
IF X > = 2 ^ 32 (overflow)
THEN saltar para DIVIDE

BYTE linhas 500 a 530

A = INT (X/256 ^ 3) byte 1
X = X * A * 256 ^ 3 resto

B = INT (X/256 ^ 2) byte 2
X = X – B * 256 ^ 2 resto
C = INT (X/256) byte 3
D = X – 256 * C byte 4

continuar com EXPOENTE
EXPOENTE linha 540
PRINT EXPOENTE = 160 – N
MANTISSA linha 550
PRINT MANTISSA = A – 128 (se S = positivo), B,C,D
saltar para SETUP para o próximo X.

O processo de expandir ou contrair a Mantissa transforma o primeiro BIT dos quatro bytes em 1. Não esquecer que estamos simular aquilo que o Interpretador da linguagem faz quando lê um número real, e nesse caso a máquina sabe exactamente o que significa activar um bit, de modo que é conveniente assinalar esse facto. Note que para números positivos, iremos fazer RESET do bit, ou seja, colocar o BIT em zero.
Na nossa simulação, fazer o Reset do bit, será equivalente a subtrair 128 (2 ^ 7) do número A. Deste modo, A será menor que 128 se o número é positivo e superior a 128 no caso inverso.
Por diferentes razões, o expoente é ajustado de modo a equilibrar com idêntica precisão, se tratamos grandes ou pequenos números. Como o expoente é sempre um inteiro, a mantissa vai ter valor a partir de D – 255; este equilíbrio é encontrado fazendo o expoente do valor absoluto de 0.5 (entrar com 1/2 no programa) igual a 128.

Teremos então:

IF ABS (X) > 1/2 THEN EXP > 128 e . . .
IF ABS (X) < 1/2 THEN EXP < 128

verifique que esta convensão dá:

X = 1/2 expoente = 128 mantissa = 0 0 0 0
X = – 1/2 expoente = 128 mantissa = 128 0 0 0

para activar o primeiro BIT dos quatro bytes, a partir de X = 1/2 teremos que incrementar N até 32. Por essa razão, quando N é subtraído desde 160 dará um expoente de 128 no «ponto de equilíbrio».
Esta polarização faz parte da convenção (tal como o bit do sinal) e pode ser usada intensamente.

FAZER O PEEK DA ÁREA DAS VARIÁVEIS

O programa 2 serve para demonstrar o facto de que inteiros e números reais são guardados de forma diversa.
O programa mostra como se guarda um número na área denominada VARS (da memória).
Ele dá os 5 bites como representação de qualquer número N, no entanto teremos de olhar para a bandeira . . . o primeiro byte será zero se o número é inteiro.
Assegure-se que o número N é a primeira variável guardada na memória, executando o comando RUN, que limpa toda a área das variáveis.
Teste, por exemplo, o número 1200 e obterá:
inteiro com 5 bytes: 0 0 176 4 0
Experimente agora o ponto flutuante usando
N = 1200.00000001

MANIPULAÇÃO DOS NÚMEROS

Será bom conhecer como ficam guardados na memória os números aos quais estão atribuídos valores «literais» dentro do programa.
Por exemplo, LET A = 5... DATA 201,193,657... IF X > 3... FOR Z = 1 TO 10.... LET R = R+1, etc.

Todos os números presentes nos enunciados descritos acima são LITERAIS.
Cada vez que um literal é lido pelo BASIC é imediatamente convertido na representação tipo inteiro ou ponto flutuante. Mais precisamente isto é executado cada vez que é encontrado um enunciado contendo literais.

```
PROGRAMA  3

  10 REM programa PEEK
  20 LET Z=10
  30 GO TO 40
  40 DATA 3,5
  50 PRINT "POS. ";TAB 12;"CODE
"
  55 LET V=PEEK 23635+256*PEEK 2
3636
  60 FOR X=V TO V+100
  70 PRINT X;TAB 10;PEEK X
  80 NEXT X
```

PEEK DA ÁREA DE PROGRAMA

O programa 3 ajuda a compreender mais alguma coisa sobre os literais fazendo o Peek da memória (extrair o valor contido numa posição da memória).
O programa analisa os primeiros 100 bytes da área de programa da memória).

1) RUN do programa, e entrega na saída até obter um CODE = 3. Isto significa ENTER e marca o final da primeira linha REM.
2) As 4 posições mais próximas contêm o número de linha e o comprimento da mesma.
3) Os 5 bytes representam as 5 teclas representativas de LET Z = 10.
4) Pode ver CODE = 14 que significa o Flag do número.
5) Os próximos 5 bytes contêm o número 10 em formato Inteiro.

VIGAS CONTÍNUAS

Autor: Fernando A. Moreira
V. N. DE GAIA

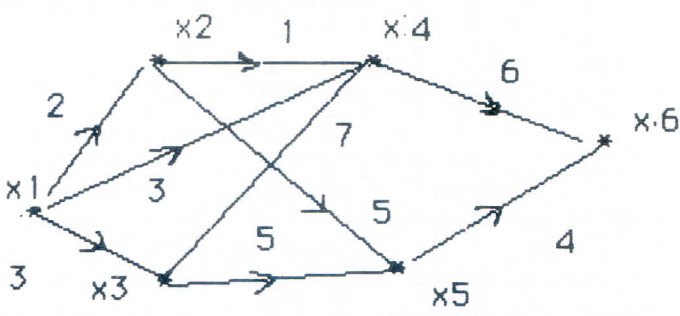
Este programa permite calcular vigas contínuas com 3 Tramos iguais.
Pode escolher uma de 5 opções de carga.
As cargas são introduzidas em Kgf e os Vãos em metros.
O programa desenha a viga com a carga escolhida e imprime as reacções, os esforços cortantes e os momentos

```
Program: vicontres
ESTE PROGRAMA PERMITE-LHE CAL
CULAR AS REACCOES NOS APOIOS
, OS ESFORCOS CORTANTES E OS
 MOMENTOS REFERENTES A QUAL
QUER VIGA CONTINUA COM TRES
 TRAMO IGUAIS EM 5 HIPOTESES
  DE CARGA


(TECLE)


 CARGA EM AB,BC E CD----TECLA 1
 CARGA EM BC -----------TECLA 2
 CARGA EM AB E BC ------TECLA 3
 CARGA EM BC E CD ------TECLA 4
 CARGA EM AB E CD-------TECLA 5

           P KG/M

A    E    B    F    C    G    D

(TECLE)

          P KG/M
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

A    E    B    F    C    G    D
CARGA  P=1500 KG/M  VAO L=8M


RA=4800  TA=4800      MA=0
         TBE=-7200    MB=-9600
RB=13200 TBD=6000     MC=-9600
         TCE=7200     MD=0
RC=13200 TCD=-6000    ME=7680
RD=4800  TD=-4800     MF=4800
                      MG=7680

    (RR E TT EM KGF
                 MM EM  KGFXM2)
```

```
   0>REM Eng. F.Antunes Moura Copyright 1985
   5 PRINT "ESTE PROGRAMA PERMITE-LHE CALCULAR AS REACCOES NOS APOIOS , OS ESFORCOS CORTANTES E OS MOMENTOS REFERENTES A QUALQUER VIGA CONTINUA COM TRES TRAMO IGUAIS EM 5 HIPOTESES DE CARGA": PRINT AT 20,0;"(TECLE)": PAUSE 0: BEEP .5,20: CLS
  50 PRINT "CARGA EM AB,BC E CD----TECLA 1"
  60 PRINT "CARGA EM BC -----------TECLA 2 "
  70 PRINT "CARGA EM AB E BC ------TECLA 3 "
  80 PRINT "CARGA EM BC E CD ------TECLA 4"
  90 PRINT "CARGA EM AB E CD-------TECLA 5"
 100 PRINT AT 12,0;"________________________________"
 120 PRINT AT 13,0;"█         █         █         █"
 125 PRINT AT 14,0;"A";AT 14,10;"B";AT 14,19;"C";AT 14,29;"D";AT 10,10;"P KG/M";AT 14,5;"E";AT 14,15;"F";AT 14,24;"G"
 128 PRINT AT 21,0;"(TECLE)"
 129 INPUT "OPCAO ";O$
 130 CLS
 140 IF O$="1" THEN GO TO 1000
 150 IF O$="2" THEN GO TO 2000
 160 IF O$="3" THEN GO TO 3000
 170 IF O$="4" THEN GO TO 4000
 180 IF O$="5" THEN GO TO 5000
 200 PAUSE 0: CLS
1010 PRINT AT 2,0;"________________________________"
1015 PRINT AT 1,0;"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
1020 PRINT AT 3,0;"█         █         █         █"
1025 PRINT AT 4,0;"A";AT 4,10;"B";AT 4,19;"C";AT 4,29;"D";AT 0,10;"P KG/M";AT 4,5;"E";AT 4,15;"F";AT 4,24;"G"
1027 INPUT "CARGA P=";P$: INPUT "VAO L=";L$
1030 PRINT AT 5,0;"CARGA  P=";P$;" KG/M";AT 5 ,20;"VAO L=";L$;"M"
1120 LET P=VAL P$: LET L=VAL L$
1200 PRINT AT 9,0;"RA=";.4*P*L;AT 9,9;"TA=";.4*P*L;AT 10,9;"TBE=";-.6*P*L;AT 11,0;"RB=";1.1*P*L;AT 11,9;"TBD=";.5*P*L;AT 13,0;"RC=";1.1*P*L;AT 12,9;"TCE=";.6*P*L;AT 13,9;"TCD=";-.5*P*L;AT 14,0;"RD=";.4*P*L;AT 14,9;"TD=";-.4*P*L;AT 9,21;"MA=0";AT 12,21;"MD=0";AT 10,21;"MB=";-.1*P*L*L;AT 11,21;"MC=";-.1*P*L*L;AT 13,21;"ME=";.08*P*L*L;AT 15,21;"MG=";.08*P*L*L;AT 14,21;"MF=";.05*P*L*L
1230 PRINT AT 20,0;"(RR E TT EM KGF
    MM EM  KGFXM2)"
1250 PAUSE 0: CLS
1300 GO TO 50
2010 PRINT AT 2,0;"________________________________"
2015 PRINT AT 1,10;"!!!!!!!!!!!"
2020 PRINT AT 3,0;"█         █         █         █"
2025 PRINT AT 4,0;"A";AT 4,10;"B";AT 4,19;"C";AT 4,29;"D";AT 0,10;"P KG/M";AT 4,5;"E";AT 4,15;"F";AT 4,24;"G"
2027 INPUT "CARGA P=";P: INPUT "VAO L=";L
2030 PRINT AT 5,0;"CARGA  P=";P$;" KG/M";AT 5 ,20;"VAO L=";L$;"M"
2200 PRINT AT 9,0;"RA=";-.5*P*L;AT 9,9;"TA=";-.05*P*L;AT 10,9;"TBE=";-.05*P*L;AT 11,0;"RB=";.55*P*L;AT 11,9;"TBD=";.50*P*L;AT 13,0;"RC=";.55*P*L;AT 12,9;"TCE=";-.5*P*L;AT 13,9;"TCD=";.05*P*L;AT 14,0;"RD=";-.05*P*L;AT 14,9;"TD=";.05*P*L;AT 9,21;"MA=0";AT 12,21;"MD=0";AT 10,21;"MB=";-.05*P*L*L;AT 11,21;"MC=";.05*P*L*L;AT 13,21;"ME=";0*P*L*L;AT 15,21;"MG=";0*P*L*L;AT 14,21;"MF=";.075*P*L*L
2230 PRINT AT 20,0;"(RR E TT EM KGF
    MM EM  KGFXM2)"
2300 PAUSE 0: CLS : GO TO 50
3010 PRINT AT 2,0;"________________________________"
3015 PRINT AT 1,0;"!!!!!!!!!!!!!!!
```

```
!!!!!!!"
3020 PRINT AT 3,0;"

3025 PRINT AT 4,0;"A";AT 4,10;"B
";AT 4,19;"C";AT 4,29;"D";AT 0,1
0;"P KG/M";AT 4,5;"E";AT 4,15;"F
";AT 4,24;"G"
3027 INPUT "CARGA P=";P: INPUT "
VAO L=";L
3030 PRINT AT 5,0;"CARGA  P=";P$
;" KG/M";AT 5 ,20;"VAO L=";L$;"M

3200 PRINT AT 9,0;"RA=";.383*P*L
;AT 9,9;"TA=";.383*P*L;AT 10,9;"
TBE=";-.617*P*L;AT 11,0;"RB=";1.
2*P*L;AT 11,9;"TBD=";.583*P*L;AT
 13,0;"RC=";.45*P*L;AT 12,9;"TCE
=";-.417*P*L;AT 13,9;"TCD=";.033
*P*L;AT 14,0;"RD=";-.033*P*L;AT
14,9;"TD=";-.033*P*L;AT 9,21;"MA
=0";AT 12,21;"MD=0";AT 10,21;"MB
=";.113*P*L*L;AT 11,21;"MC=";-.0
33*P*L*L;AT 13,21;"ME=";.074*P*L
*L;AT 15,21;"MG=";0*P*L*L;AT 14,
21;"MF=";.34*P*L*L
3230 PRINT AT 20,0;"(RR E TT EM
KGF
  MM EM  KGFXM2)"
3300 PAUSE 0: CLS : GO TO 50
4010 PRINT AT 2,0;"________________
________________
4015 PRINT AT 1,10;"!!!!!!!!!!!!
!!!!!!!!"
4020 PRINT AT 3,0;"

4025 PRINT AT 4,0;"A";AT 4,10;"B
";AT 4,19;"C";AT 4,29;"D";AT 0,1
0;"P KG/M";AT 4,5;"E";AT 4,15;"F
";AT 4,24;"G"
4027 INPUT "CARGA P=";P: INPUT "
VAO L=";L
4030 PRINT AT 5,0;"CARGA  P=";P$
;" KG/M";AT 5 ,20;"VAO L=";L$;"M

4200 PRINT AT 9,0;"RA=";-.033*P*
L;AT 9,9;"TA=";-.033*P*L;AT 10,9
;"TBE=";.033*P*L;AT 11,0;"RB=";.
45*P*L;AT 11,9;"TBD=";-.417*P*L;
AT 13,0;"RC=";1.2*P*L;AT 12,9;"T
CE=";.583*P*L;AT 13,9;"TCD=";-.6
17*P*L;AT 14,0;"RD=";.383*P*L;AT
 14,9;"TD=";.383*P*L;AT 9,21;"MA
=0";AT 12,21;"MD=0";AT 10,21;"MB
=";0*P*L*L;AT 11,21;"MC=";.34*P*
L*L;AT 13,21;"ME=";.074*P*L*L;AT
 15,21;"MG=";.113*P*L*L;AT 14,21
;"MF=";-.033*P*L*L
4230 PRINT AT 20,0;"(RR E TT EM
KGF
  MM EM  KGFXM2)"
4300 PAUSE 0: CLS : GO TO 50
5010 PRINT AT 2,0;"________________
________________
5015 PRINT AT 1,0;"!!!!!!!!!!!";A
T 1,20;"!!!!!!!!!!"
5020 PRINT AT 3,0;"

5025 PRINT AT 4,0;"A";AT 4,10;"B
";AT 4,19;"C";AT 4,29;"D";AT 0,1
0;"P KG/M";AT 4,5;"E";AT 4,15;"F
";AT 4,24;"G"
5027 INPUT "CARGA P=";P: INPUT "
VAO L=";L
5030 PRINT AT 5,0;"CARGA  P=";P$
;" KG/M";AT 5 ,20;"VAO L=";L$;"M

5200 PRINT AT 9,0;"RA=";.45*P*L;
AT 9,9;"TA=";.45*P*L;AT 10,9;"TB
E=";-.5*P*L;AT 11,0;"RB=";.55*P*
L;AT 11,9;"TBD=";.05*P*L;AT 13,0
;"RC=";.55*P*L;AT 12,9;"TCE=";.0
5*P*L;AT 13,9;"TCD=";-.05*P*L;AT
 14,0;"RD=";.45*P*L;AT 14,9;"TD=
";.45*P*L;AT 9,21;"MA=0";AT 12,2
1;"MD=0";AT 10,21;"MB=";-.05*P*L
*L;AT 11,21;"MC=";-.05*P*L*L;AT
13,21;"ME=";.102*P*L*L;AT 15,21;
"MG=";.102*P*L*L;AT 14,21;"MF=";
0*P*L*L
5230 PRINT AT 20,0;"(RR E TT EM
KGF
  MM EM  KGFXM2)"
5300 PAUSE 0: CLS : GO TO 50
```

Aplicação da Teoria dos GRAFOS À TOPOLOGIA URBANÍSTICA

Apresentamos um programa capaz de calcular o comprimento mínimo de cada um dos pontos de um «GRAFO» relativamente a um ponto de referência. Este tipo de problemas tem grande aplicação, quer na resolução dos problemas de cálculo de custos de transportes, quer no fixar de itinerário óptimos.

```
VALOR DE CADA'NO' :
X1=0
X2=2
X3=3
X4=3
Valor do percurso minimo
 de X1 a X4:3

VALOR DE CADA'NO' :
X1=0
X2=2
X3=4
X4=9
Valor do percurso maximo
 de X1 a X4:9
```

```
  1 BORDER 0: PAPER 0: INK 7
 10 REM calculo de percurso min
imo ou maximo
 20 REM entre dois pontos de um
grafo
 30 REM inicializar
 50 INPUT "Numero de 'nos' do g
rafo ? ";n
 60 DIM X(n): DIM T(n,n)
 70 PRINT "Seleccionar:"
 74 PRINT : PRINT "1 - Percurso
minimo": PRINT : PRINT "2 - Per
curso maximo"
 80 PRINT AT 19,0;"": INPUT "nu
mero ? ";RE
 90 IF RE<1 OR RE>2 THEN GO TO
80
100 IF RE=1 THEN GO TO 110
105 GO TO 160
110 LET X(1)=0
120 FOR I=2 TO N
130 LET X(I)=10*EXP 10
140 NEXT I
150 GO TO 200
160 REM atribuir zeroa cada no
170 FOR I=1 TO N
180 LET X(I)=0
190 NEXT I
200 REM VALOR DE CADA ARCO
210 FOR I=1 TO N-1
220 FOR J=2 TO N
```

```
230 READ T(I,J)
240 NEXT J
250 NEXT I
260 REM TRATAMENTO
270 FOR I=1 TO N-1
280 FOR J=2 TO N
290 IF T(I,J)=0 THEN GO TO 360
300 IF RE=1 THEN GO TO 310
305 GO TO 320
310 IF X(J)-X(I)<=T(I,J) THEN G
O TO 360
315 GO TO 330
320 IF X(J)-X(I)>=T(I,J) THEN G
O TO 360
330 LET X(J)=X(I)+T(I,J)
340 IF I<=J THEN GO TO 360
350 LET I=J: IF RE=1 THEN LET J
=0
355 IF RE<>1 THEN LET J=2
360 NEXT J
370 IF RE=1 THEN LET J=0
375 IF RE<>1 THEN LET J=2
380 NEXT I
390 REM RESULTADOS
400 CLS : PRINT "VALOR DE CADA'
NO' : "
410 FOR I=1 TO N
420 PRINT "X";I;"=";X(I)
430 NEXT I
440 IF RE=1 THEN GO TO 460
450 PRINT "Valor do percurso ma
ximo": PRINT " de X1 a X";n;":";
X(n): STOP
460 PRINT "Valor do percurso mi
nimo": PRINT " de X1 a X";n;":";
X(n): STOP
470 REM entrada dos dados
480 DATA 2,3,3,0,0
490 DATA 0,0,1,5,0
500 DATA 0,0,7,5,0
510 DATA 0,0,0,0,6
520 DATA 0,0,0,0,4
```

ORIENTAÇÃO DE ANTENAS

Autor: Paulo Metelo
ODIVELAS

Este programa é dedicado aos radioamadores que tenham antenas direccionais e que queiram orientar a sua antena através de coordenadas.

O programa fornece a direcção do (QTH alvo) a partir do (QTH local), pela via curta (short path) e pela via longa (long path), isto tudo de 0 a 360° sempre no sentido dos ponteiros do relógio, a partir do norte verdadeiro.

Além disto tudo, diz-nos as distâncias entre os dois pontos para ambas as vias (curta e longa) em quilómetros.

As coordenadas dos dois locais são introduzidas da seguinte forma: aaa.aab ou seja: exemplo — latitude de 25° 45′ Sul vamos introduzir 025.45S. Se a longitude for 005° 39′ Oeste vamos introduzir 005.39W.

NOTA: É muito importante que se usem as letras «S» e «E» para Sul e Este respectivamente. Pois sempre que a latitude for Sul e a longitude for Este o sinal dos dados é negativo (linhas 55,80,125 e 175).

Sempre que a longitude do alvo e a longitude do local forem iguais o cálculo será impossível mas isto é remediado pela (linha 173) pois é somado 1.10-9 rad. à longitude do alvo. É possível calcular a direcção de qualquer alvo distante de 1 minuto de arco, cerca de 1,8 km do QTH local.

Para saber as coordenadas a introduzir deve consultar um Atlas, pois geralmente estes trazem as coordenadas e assim será simples de as saber.

Linhas (35,60,95 e 140) Transforma graus e minutos em radianos

Linhas (50,75,120 e 170)

Linhas (55,80,125 e 175) Sempre que as latitudes sejam Sul e as longitudes Este, o sinal dos dados é negativo.

Linha (122) Para determinar a direcção dos alvos sobre a linha do equador.

Linha (173) Torna o cálculo possível quando a longitude do local é igual à longitude do alvo.

Linha (247) Torna o cálculo impossível.

```
1 REM orientacao de antenas
2 REM programa adaptado da re
vista ANTENA ELECTRONICA POPULAR
para o spectrum
4 BORDER 7: PAPER 6: INK 0: C
LS
5 PRINT AT 0,0;"EXEMPLO: para
a lat. 22g 50m sul = 022.50S
para a long. 1g 30m oeste
= 001.30W e impor-tante utiliza
r as letras ""S"" e ""E"" em ma
iusculas para sul e este resp
ectivamente. E conveni
ente que a orientacao venha em
maiusculas:(N,S,E,W)."
6 PAUSE 800: CLS
7 LET x=PI/180
10 LET y=PI/10800
20 LET a$="?          "
25 LET b$="?          "
30 PRINT AT 0,5;;"ORIENTACAO D
E ANTENAS";AT 2,3;"lat. local";A
T 2,15;a$
35 IF a$(1)="?" THEN INPUT a$:
IF LEN a$<>7 THEN GO TO 20
40 IF a$="" THEN GO TO 20
45 PRINT AT 2,15;a$;AT 3,3;"lo
ng.local";AT 3,15;b$;AT 19,0;"SE
DADOS INCORRECTOS: ""E"""
50 LET lal=VAL a$( TO 3)*x+VAL
a$(5 TO 6)*y
55 IF a$(7)="S" THEN LET lal=-
lal
60 IF b$(1)="?" THEN INPUT b$
65 IF b$="" THEN GO TO 25
70 IF b$="E" THEN GO TO 20
73 IF LEN b$<>7 THEN GO TO 25
75 LET lnl=VAL b$( TO 3)*x+VAL
b$(5 TO 6)*y
80 IF b$(7)="E" THEN LET lnl=-
lnl
85 LET c$="?          "
90 PRINT AT 3,15;b$;AT 5,3;"la
t. alvo";AT 5,15;c$
95 INPUT c$
100 IF c$="" THEN GO TO 95
105 IF c$="E" THEN GO TO 25
110 IF LEN c$<>7 THEN GO TO 95
120 LET laa=VAL c$( TO 3)*x+VAL
c$(5 TO 6)*y
122 IF laa=0 THEN LET laa=1E-9
125 IF c$(7)="S" THEN LET laa=-
laa
130 LET d$="?          "
135 PRINT AT 5,15;c$;AT 6,3;"lo
ng.alvo";AT 6,15;d$
140 INPUT d$
145 IF d$="" THEN GO TO 140
150 IF d$="E" THEN GO TO 85
155 IF LEN d$<>7 THEN GO TO 130
160 PRINT AT 6,15;d$;AT 21,0;"S
E DADOS CORRECTOS: ""ENTER"""
170 LET lna=VAL d$( TO 3)*x+VAL
d$(5 TO 6)*y
173 IF lna=lnl THEN LET lna=lna
+1E-9
175 IF d$(7)="E" THEN LET lna=-
lna
180 INPUT u$
185 IF u$="E" THEN GO TO 130
186 BEEP .5,30: BEEP .8,20: BEE
```

```
P .3,15
 190 FOR j=0 TO 31
 195 PRINT AT 7,j;CHR$ 42
 200 NEXT j
 210 PRINT AT 19,0;"          ";
AT 19,10;"CALCULANDO";AT 19,20;"
"
 215 PRINT AT 21,0;"
"
 220 LET l=lnl-lna
 225 IF l<-PI THEN LET l=l+2*PI
 230 IF l>PI THEN LET l=l-2*PI
 245 LET f=ATN (COS l*1/TAN laa)
 247 IF COS (lal+f)=0 THEN GO TO
 400
 250 LET c=ATN (TAN l*SIN f/COS
(lal+f))
 255 IF l>0 AND c<0 THEN LET c=c
+PI
 260 IF l<0 AND c>0 THEN LET c=c
+PI
 265 IF l<0 AND c<0 THEN LET c=c
+2*PI
 270 LET c=c*1/x
 275 LET d=ACS (SIN lal*SIN laa+
COS lal*COS laa*COS l)
 280 PRINT AT 9,2;"direccao:";AT
 11,3;"via curta= ";INT c;" G ";
INT ((c-INT c)*60*1)/1;" MIN   "
 285 IF c>180 THEN LET cl=c-180
 290 IF c<=180 THEN LET cl=c+180
 295 PRINT AT 12,3;"via longa= "
;INT cl;" G ";INT ((cl-INT cl)*6
0*1)/1;" MIN    "
 300 LET dkm=INT (6366.1977*d*10
)/10
 310 PRINT AT 14,2;"distancia:";
AT 16,3;"via curta = ";dkm;" KM
"
 313 PRINT AT 17,3;"via longa =
";40000-dkm;" KM    "
 315 FOR j=0 TO 31
 320 PRINT AT 19,j;CHR$ 42
 325 NEXT j
 330 PRINT AT 20,0;"para novo al
vo:""ENTER"" ";AT 21,0;"para nov
o QTH local:  ""N"""
 335 INPUT u$
 340 CLS
 345 IF u$="N" OR u$="n" THEN GO
 TO 7
 350 GO TO 30
 400 PRINT AT 12,6;"calculo impo
ssivel"
 415 GO TO 315
```

DESENHO DO MOCHO

R. C./VILA NOVA DE GAIA

O COELHINHO DA PLAY BOY

```
  5 REM BUNNIE - 23FEV85
 10 CLS : PLOT 100,2
 20 DRAW 0,16
 30 DRAW 10,-6
 40 DRAW 10,4
 50 DRAW 0,-16
 60 DRAW -10,6
 70 DRAW -10,-4
100 PLOT 108,14
110 DRAW 0,16,-PI/6
120 DRAW -24,16,-PI/3
130 DRAW 0,24,-PI/3
140 DRAW 50,24,-PI/3
150 DRAW -64,80,-PI/2.5
160 DRAW 20,-18,PI/12
170 DRAW 52,-62,-PI/6
180 DRAW 24,80,-PI/3
190 DRAW -16,-82,-PI/3
200 DRAW 8,-70,-PI/2
210 DRAW -34,-8,-PI/12
220 DRAW 0,8
230 DRAW -18,-8
300 PLOT 124,4
310 DRAW 0,6
320 DRAW 32,6
330 DRAW 0,-6
340 DRAW -32,-6
400 CIRCLE 108,72,4
500 STOP
```

AOS SÓCIOS

Muitos jogos gravados em cassete nos têm sido enviados por alguns de vocês, por troca ou simples oferta. Embora tenhamos uma enorme lista de programas sem instruções desejaríamos o seguinte:
Sempre que nos enviem programas, façam o possível por enviar as instruções ou mesmo um descritivo que facilite a sua utilização.
A seguir enumeramos uma lista de alguns dos jogos que possuímos, mas sem quaisquer instruções.

White Ligthning	Magic Mountain
Space Station Zebra	1984
Jumbly	1994
Velnor's Lair	Magic Meanies
Bedlam	Voyage Into The Unknown
Groucho	

Agradecemos que, se conhece algum deles e se lhe for possível, nos envie um descritivo ou uma cópia das instruções originais.

CÁLCULO DE RAÍZES

Autor: Pedro Azevedo

Foi publicado na revista de Dezembro de 84, um programa para cálculo de raízes de índice superior a 2, por Carlos Moreno para o Spectrum. É dito junto ao programa que o Spectrum não tira raízes de índice superior a 2, o que é verdade, mas como um número elevado a uma potência fraccionária é igual à raiz, sendo o índice o denominador da potência, desse número elevado à potência do numerador, isto é:

$$B^{c/a} = \sqrt[r]{B^{c}}$$

Este caso o Spectrum já executa sem problemas e com a mesma rapidez que executa qualquer outra operação.

Não há qualquer intenção de tirar o valor do programa apresentado por Carlos Moreno, que dá mais 2 casas decimais que o Spectrum, mas que para aplicações práticas não são necessárias, chegando o valor apresentado pelo Spectrum, introduzindo o seguinte programa:

```
10 PRINT AT 0,0;«A»:PLOT 0,166:DRAW 10,0:DRAW 5,-
   20:DRAW 5,25:DRAW 50,0:
                    PRINT AT 2,5;«B»
20 INPUT «Indice da raiz»;A:INPUT «Radicando»;B:INPUT
   «Elevado a potência»,C
30 PRINT AT 0,0;A;AT 2,5;B↑C:PRINT AT 2,9;«=»:PLOT
   0,166:DRAW 10,0:DRAW 5,—20:
                    DRAW 5,25:DRAW 50,0
40 PRINT AT 2,10;B↑(C/A)
50 FOR N=0 TO 500:NEXT N
60 PAUSE 0:GOTO 10
```

NOTA: Por exemplo, quando se quer calcular a raíz cúbica de 27, com este programa dá-se os seguintes valores:

Índice da raíz=3
Radicando=27
Elevado à potência=1

Este programa pode ainda ser melhorado com as seguintes linhas, o que evita que dê condições «Matematicamente impossíveis» e as habituais mensagens de erro:

```
21 IF A=0 THEN GOTO 20
22 IF ((INT (A/2)) <> (A/2)) AND (((ABS b) <> b AND ((INT
(C/2)) <> ((C/2)) THEN GOTO 20
```

PROGRAMA CÁLCULO DE TRANSFORMADORES

Autor: Paulo Metelo
ODIVELAS

Este programa é dedicado àqueles que gostam de fazer os seus próprios transformadores de alimentação.

Com a simples introdução de alguns dados ele vai-nos dar o número de espiras do primário e dos diversos secundários, para além do diâmetro e secção dos seus fios, dá-nos também a secção do transformador, a potência total e a potência individual dos secundários.

Este programa no cálculo prático está concebido para uma frequência de rede de 50 Hz.

Existem duas possibilidades de cálculo, uma prática e uma teórica, se quisermos dá-nos também uma tabela de fios de cobre que tem uma gama compreendida entre 5^{mm} e $6{,}5^{mm}$ para além da secção e da resistência em ohms/km ou ohms/m.

Tem ainda a possibilidade de escolher entre três qualidades de núcleos (lâminas comuns, de silício e núcleo de ferrite).

Vão ser pedidas por duas vezes as tensões secundárias e suas correntes, a primeira é empregue no cálculo das potências a segunda no cálculo das espiras, diâmetros e secções do secundário, portanto das duas instruções de dados, os segundos deverão ser iguais aos primeiros.

E para terminar muito simplesmente, aqui estão as partes em que se divide o programa:

Linhas 1 a 7=instruções de funcionamento
Linhas 8 a 165=cálculo prático
Linhas 1000 a 1049=cálculo teórico
Linhas 2000 a 2045=tabela de fios de cobre

```
   1 REM calculo de transformado
res
   2 PRINT AT 3,0;"*************
******************": PRINT AT 4,
0;"*
  *"
   3 PRINT AT 5,0;"*  CALCULO DE
 TRANSFORMADORES *": PRINT AT 6,
0;"*
  *"
   4 PRINT AT 7,0;"*************
******************"
   5 PRINT AT 9,0;"Este programa
 calcula a potencia individual
e  total  dos  enr-rolamentos a
seccao do nucleo,  as espiras  d
o secundario e do  primario, os
diametros e seccoes dos seus res
pectivos fios."
   6 PRINT "Alem disto da-nos a
escolha um  calculo pratico ou t
eorico para alem de uma tabela d
e fios de    cobre.
              Os calculos praticos
 sao feitos para uma frequencia
de rede de  50 Hz."
   7 PAUSE 300: CLS
   8 CLS : REM Calculo pratico
   9 PRINT AT 10,3;"Quer um calc
ulo teorico ou  ": PRINT AT 12,3
;"pratico? (t/p)
  "
  10 INPUT a$
  11 IF a$="p" OR a$="P" THEN GO
 TO 18
  12 IF a$="t" OR a$="T" THEN GO
 TO 1000
  13 PRINT AT 10,3;"Erro na intr
oducao da opccao ": PRINT AT 12,
3;"Introduza novamente a opccao.
"
  14 PAUSE 100: GO TO 8
  18 CLS : PRINT AT 4,2;"Quadro
da inducao magnetica em  Weber/m
2": PRINT AT 7,0;"Para laminas c
omuns      1   W/m2": PRINT AT 9,
0;"Para laminas de silicio 1.2 W
/m2": PRINT AT 11,0;"Para nucleo
 de ferrite 0.34 W/m2"
  19 LET pt=0
  20 INPUT "tensao do primario e
```

```
m volts ";vp
  30 INPUT "inducao magnetica em
 weber/m2 ";b
  35 CLS
  40 INPUT "quantos enrrolamento
s tem o sec. ";x
  50 INPUT "tensao individual do
 secundario em volts ";v
  55 INPUT "corrente individual
do secunda. em amperes  ";i
  58 LET n=x-1
  60 LET po=v*i: PRINT "pot. do
enrrola. ";n;" em W = ";po
  65 LET pt=pt+po
  70 LET x=x-1
  75 IF x<>0 THEN GO TO 50
  80 PRINT "pot. de todos enr. e
m W = ";pt
  85 LET s=1.35*SQR pt: PRINT "s
ec. do transf. em Cm2 = ";s
  90 INPUT "quantos enrrolamento
s tem o sec. ";x
  95 INPUT "tensao individual do
 secundario em volts ";v
   0>INPUT "corrente individual
do secunda. em amperes ";i
 102 LET z=x-1
 103 PRINT "
 "
 105 LET es=(45.9/s*b)*v: PRINT
"espiras do secunda.";z;"= ";es
 110 LET ds=0.7*SQR i: PRINT "di
a. do sec.";z;" em mm = ";ds
 115 LET se=0.785*ds↑2: PRINT "s
ec. do sec.";z;" em Cm2= ";se
 120 LET x=x-1
 125 IF x<>0 THEN GO TO 95
 126 PRINT "
 "
 135 LET pri=(44.1/(s*b))*vp: PR
INT "espiras do primario = ";pri
 140 LET dp=0.7*SQR (pt/vp): PRI
NT "dia. fio do pri. mm = ";dp
 143 LET sp=0.785*dp↑2: PRINT "s
ec. fio do pri. Cm2= ";sp
 150 PAUSE 4000: CLS
 156 PRINT AT 10,3;"Quer uma tab
ela de          ": PRINT AT 12,
3;"fios de cobre? (s/n)
        ": INPUT b$
 158 IF b$="n" OR b$="N" THEN GO
 TO 8
 159 IF b$="s" OR b$="S" THEN GO
 TO 2000
 160 PRINT AT 10,3;"Erro na intr
oducao da opccao": PRINT AT 12,3
;"Introduza novamente a opccao."
 165 PAUSE 100: GO TO 156
1000 REM Calculo teorico
1001 CLS : PRINT AT 4,2;"Quadro
da inducao magnetica em  Weber/m
2": PRINT AT 7,0;"Para laminas c
omuns         1 W/m2": PRINT AT 9,
0;"Para laminas de silicio 1.2 W
/m2": PRINT AT 11,0;"Para nucleo
 de ferrite 0.34 W/m2"
1002 LET pt=0
1004 INPUT "tensao do primario e
m volts ";vp
1005 INPUT "frequencia da rede e
m Hz ";F
1006 INPUT "inducao magnetica em
 Weber/m2 ";b
1007 CLS
1009 INPUT "quantos enrrolamento
s tem o sec. ";x
1011 INPUT "tensao individual do
 secundario em volts ";v
1013 INPUT "corrente individual
do secunda. em amperes ";i
1014 LET n=x-1
1015 LET po=v*i: PRINT "pot. do
enrrola. ";n;" em W =";po
1017 LET pt=pt+po
1019 LET x=x-1
1021 IF x<>0 THEN GO TO 1011
1023 PRINT "pot. de todos enr. e
m W=";pt
1025 LET s=1.5*SQR pt: PRINT "se
c. do transf. em Cm2 =";s
1026 PRINT "
 "
1027 LET pri=(10000*vp/(4.44*F*s
*b)): PRINT "espiras do primario
 =";pri
1029 LET dp=0.7*SQR (pt/vp): PRI
NT "dia. fio do pri. mm =";dp
1031 LET sp=0.785*dp↑2: PRINT "s
ec. fio do pri. Cm2=";sp
1033 INPUT "quantos enrrolamento
s tem o sec. ";x
1035 INPUT "tensao individual do
 secundario em volts ";v
1037 INPUT "corrente individual
do secunda. em amperes ";i
1038 LET z=x-1
1039 PRINT "
 "
1040 LET es=(v/vp)*pri: PRINT "e
spiras do secunda.";z;"= ";es
1041 LET ds=0.7*SQR i: PRINT "di
a. do sec.";z;" em mm =";ds
1043 LET se=0.785*ds↑2: PRINT "s
ec. do sec.";z;" em Cm2=";se
1045 LET x=x-1
1047 IF x<>0 THEN GO TO 1035
1049 GO TO 144
2000 REM tabela de fios de cobre
2001 CLS
2002 PRINT AT 3,0;"*************
*******************": PRINT AT 4,
0;"*
 *": PRINT AT 5,0;"*       TABEL
A DE FIOS        * ": PRINT AT 6
,0;"*           DE COBRE
 *": PRINT AT 7,0;"*
               *": PRINT AT 8
,0;"****************************
***"
2004 PAUSE 150: CLS
2006 PRINT AT 0,0;"Dia. mm   Sec
. Cm2    Resi. ohms"
2007 PRINT "____________________
________"
2008 PRINT ".05        .00196
  8.91     .06         .00283
  6.19     .07         .00385
  4.55     .08         .00503
  3.48     .09         .00636
  2.75     .10         .00785
  2.23     .12         .0113
  1.55     .14         .0154
  1.137    .15         .0177
  .990"
2010 PRINT ".16        .0201
  .870     .18         .0254
  .688     .20         .0314
  .557     .22         .0380
  .460     .25         .0491
  .356     .28         .0616
  .284     .30         .0707
  .248     .32         .0804
  .218     .35         .0962
  .182     .38         .1134
  .154     .40         .1257
  .139"
2015 PRINT ".42        .1385
  .126     .45         .1590
  .110     .48         .1810
  .0967    .50         .1963
  .0891    .52         .2124
  .0824    .55         .2376
  .0737    .58         .2642
  .0662    .60         .2827
  .0619    .62         .3019
  .0580    .65         .3318
  .0527"
2020 PRINT ".68        .3632
  .0482    .70         .3848
  .0455    .75         .4418
  .0396    .80         .5027
  .0348    .85         .5675
  .0308    .90         .6362
  .0275    .95         .7088
  .0247    1.00        .7854
  .0223    1.05        .8659
  .0202    1.10        .9503
  .0184    1.15       1.0387
```

```
   .0168"
2025 PRINT "1.20          1.131
   15.47    1.25          1.227
   14.26    1.30          1.327
   13.18    1.35          1.431
   12.23    1.40          1.539
   11.37    1.50          1.767
   9.903    1.60          2.011
   8.704    1.70          2.270
   7.710    1.80          2.545
   6.877    1.90          2.835
   6.172"
2030 PRINT "2.00          3.142
   5.570    2.1           3.464
   5.053    2.2           3.801
   4.604    2.3           4.155
   4.212    2.4           4.524
   3.868    2.5           4.909
   3.565    2.6           5.309
   3.296    2.7           5.726
   3.056    2.8           6.158
   2.842    2.9           6.605
   2.649"
2035 PRINT "3.0           7.069
   2.476    3.1           7.548
   2.319    3.2           8.042
   2.176    3.3           8.553
   2.046    3.4           9.079
   1.927    3.5           9.621
   1.819    3.6           10.179
   1.719    3.7           10.752
   1.628    3.8           11.341
   1.543    3.9           11.946
   1.465"
2040 PRINT "4.0           12.566
   1.393    4.2           13.854
   1.263    4.4           15.205
   1.151    4.5           15.904
   1.100    4.6           16.619
   1.053    4.8           18.096
   0.967    5.0           19.635
   0.891    5.5           23.758
   0.737    5.8           26.421
   0.662    6.0           28.274
   0.619    6.5           33.183
   0.527"
2045 PAUSE 150: CLS
2046 PRINT AT 5,3;"Quer imprimir
 ou            ": PRINT AT 7,3
;"voltar ao menu ?         ": PRIN
T AT 9,11;"(i/m)"
2047 INPUT a$
2048 IF a$="i" OR a$="I" THEN GO
 TO 2051
2049 IF a$="m" OR a$="M" THEN GO
 TO 8
2050 PRINT AT 5,3;"Erro na intro
ducao da opccao.               ":
 PRINT AT 7,3;"Introduza nova op
ccao.       ": PRINT AT 9,0;"
                               ": PAU
SE 100: GO TO 2046
2051 LPRINT "transfor"
2052 LLIST
9998 STOP
9999 SAVE "transfor" LINE 1: PRI
NT "VERIFY": VERIFY "transfor":
CLS : PRINT "Esta bem gravado"
```

MOZART C 1984

Adapt.: por Roseira Coelho
V. N. DE GAIA

Chi-Yeung Choy de Ruislip dobra os seus dedos e realiza esta obra-prima musical.

Até à bem pouco tempo o SPECTRUM era considerado pouco efectivo em qualquer aplicação musical razoável. Para lhe provarmos o contrário apresentamos-lhe um programa que o fará reconsiderar a sua posição.

Em vez da selecção usual de efeitos de som ou de breves melodias conhecidas Chi-Yeung programou, com sucesso, uma versão para computador do primeiro movimento da sonata de MOZART para piano em C Maior K 545.

O que é mais espantoso é que foi todo escrito em Basic e funciona em 16 K.

Foi dada uma atenção especial a todos os detalhes técnicos. Dê aos seus dedos descanso e prazer aos seus ouvidos. Comece Maestro!

```
   5 REM SONATA PARA PIANO K545
            DE MOZART
  10 CLS : GO TO 3000
  50 READ B,C,D
  60 BEEP 2*X,B: BEEP X,C: BEEP
X,D: RETURN
  70 FOR M=1 TO 16
  80 READ B: BEEP Z,B
  90 NEXT M: RETURN
 100 FOR N=1 TO 8
 110 READ B: BEEP Y,B
 120 NEXT N
 130 BEEP X,14: BEEP X,19: BEEP
X,7: PAUSE 25
 140 RETURN
 150 READ B,C,D,E,F
 160 BEEP 2*X,B: BEEP Z,C: BEEP
Y+Z,D: BEEP Z,E: BEEP Y+Z,F: RET
URN
 170 FOR N=1 TO 12
 180 READ B: BEEP Z,B
 190 NEXT N: RETURN
 200 BEEP X,19
 210 FOR N=1 TO 12
 220 READ B: BEEP Z,B-12
 230 NEXT N: RETURN
 240 READ B,C,D
 250 BEEP 2*X,B+5: BEEP X,C+5: B
EEP X,D+5: RETURN
 530 LET X=1/2: LET Y=X/2: LET Z
=X/4: LET A=X/8
 540 GO SUB 50
 550 READ B,C,D,E
 560 BEEP X+Y,B: BEEP Z,C: BEEP
Z,D: BEEP X,E: PAUSE 25
 570 GO SUB 50
 580 BEEP X,19
 590 FOR N=1 TO 3
 600 BEEP A,19: BEEP A,17: NEXT
N
 610 BEEP A,16: BEEP A,17: BEEP
X,16: PAUSE 25
 620 FOR N=1 TO 5
 630 READ B: BEEP Y,B
 640 FOR M=1 TO 14
 650 READ B: BEEP Z,B: NEXT M: N
EXT N
 660 GO SUB 70
 670 GO SUB 100
 680 FOR N=1 TO 2
 690 FOR M=1 TO 4
 700 IF N=1 THEN BEEP Z,1: BEEP
Z,2: GO TO 720
 710 BEEP Z,0: BEEP Z,2
 720 NEXT M: NEXT N
 730 RESTORE 2120
 740 FOR N=1 TO 2
 750 RESTORE 2120
 760 READ B,C,D,E,F,G,H,I,J,K,L,
M,P
 770 BEEP Y,B: BEEP Y,C: BEEP X+
Y,D: BEEP Z,E: BEEP Z,F: BEEP Y,
G: BEEP Y,H: BEEP A,I: BEEP A,J:
 BEEP A,K: BEEP A+Z,L: BEEP Z,M:
 BEEP X,P: PAUSE 50
 780 NEXT N
 790 FOR N=1 TO 4
 800 GO SUB 70: NEXT N
```

```
 810 GO SUB 150
 820 BEEP Z,20: BEEP X+Z,21: BEE
P A,23: BEEP A,21: BEEP A,20: BE
EP A,21: BEEP Y,24: BEEP Y,21: B
EEP Y,24: BEEP Y,21
 830 BEEP Y,23: BEEP Y,19: BEEP
2*X,26: BEEP Z,24: BEEP Z,23: BE
EP Z,21: BEEP Z,19
 840 FOR N=1 TO 15
 850 BEEP A,23: BEEP A,21: NEXT
N
 860 BEEP Z,19: BEEP Z,21: BEEP
X,19
 870 GO SUB 170
 880 RESTORE 2180: GO SUB 200
 890 BEEP X,7: BEEP X,23: BEEP X
,19: PAUSE 25
 900 BEEP X,7: GO SUB 170
 910 RESTORE 2190: GO SUB 200
 920 FOR N=1 TO 2
 930 GO SUB 70: NEXT N
 940 BEEP X,17
 950 RESTORE 2190
 960 FOR N=1 TO 12
 970 READ B: BEEP Z,B-5: NEXT N
 980 BEEP X,14
 990 RESTORE 2190
1000 FOR N=1 TO 12
1010 READ B: BEEP Z,B-17: NEXT N
1020 RESTORE 2220
1030 FOR N=1 TO 7
1040 GO SUB 70: NEXT N
1050 RESTORE 2030: GO SUB 240
1060 READ B,C,D,E
1070 BEEP X+Y,B+5: BEEP Z,C+5: B
EEP Z,D+5: BEEP X,E+5: PAUSE 25
1080 GO SUB 240
1090 BEEP X,24
1100 FOR N=1 TO 3
1110 BEEP A,24: BEEP A,22: NEXT
N
1120 BEEP A,21: BEEP A,22: BEEP
X,21: PAUSE 25
1130 FOR N=1 TO 4
1140 READ B: BEEP Y,B+5
1150 FOR M=1 TO 14
1160 READ B: BEEP Z,B+5
1170 NEXT M: NEXT N
1180 BEEP 2*X,21: PAUSE 25: BEEP
 X,21
1190 BEEP 2*X,19: PAUSE 25: BEEP
 X,19
1200 BEEP 2*X,17: PAUSE 25: BEEP
 X,17
1210 BEEP 2*X,16: PAUSE 25: BEEP
 X,16
1220 RESTORE 2300
1230 GO SUB 70
1240 RESTORE 2100
1250 GO SUB 70: GO SUB 100
1260 FOR N=1 TO 2
1270 FOR M=1 TO 16
1280 IF N=2 THEN BEEP Z,5: BEEP
Z,7: GO TO 1300
1290 BEEP Z,6: BEEP Z,7
1300 NEXT M: NEXT N
1310 FOR N=1 TO 2
1320 RESTORE 2120
1330 READ B,C,D,E,F,G,H,I,J,K,L,
M,P
1340 BEEP Y,B-7: BEEP Y,C-7: BEE
P X+Y,D-7: BEEP Z,E-7: BEEP Z,F-
7: BEEP Y,G-7: BEEP Y,H-7: BEEP
A,I-7: BEEP A,J-7: BEEP A,K-7: B
EEP A+Z,L-7: BEEP Z,M-7: BEEP X,
P-7: PAUSE 50
1350 NEXT N
1360 FOR N=1 TO 2
1370 FOR M=1 TO 16
1380 READ B: BEEP Z,B-7
1390 NEXT M: NEXT N
1400 FOR N=1 TO 2
1410 FOR M=1 TO 16
1420 READ B: BEEP Z,B+5
1430 NEXT M: NEXT N
1440 RESTORE 2290
1450 GO SUB 150
1460 BEEP 2*X,21: BEEP Z,20: BEE
P Y+Z,21: BEEP Z,20: BEEP Y+Z,21
1470 BEEP Y,19
1480 RESTORE 2310
1490 FOR N=1 TO 14
1500 READ B: BEEP Z,B: NEXT N
1510 FOR N=1 TO 15
1520 BEEP A,16: BEEP A,14: NEXT
N
1530 BEEP Z,12: BEEP Z,14: BEEP
X,12
1540 RESTORE 2180
1550 FOR N=1 TO 12
1560 READ B: BEEP Z,B-7: NEXT N
1570 BEEP X,12
1580 RESTORE 2180
1590 FOR N=1 TO 12
1600 READ B: BEEP Z,B-19: NEXT N
1610 BEEP X,0: BEEP X,16: BEEP X
,12: PAUSE 25
2030 DATA 12,16,19,11,12,14,12
2040 DATA 21,19,24
2050 DATA 9,11,12,14,16,17,19,21
,19,17,16,14,12,11,9
2060 DATA 7,9,11,12,14,16,17,19,
17,16,14,12,11,9,7
2070 DATA 5,7,9,11,12,14,16,17,1
6,14,12,11,9,7,5
2080 DATA 4,5,7,9,11,12,14,16,14
,12,11,9,7,5,4
2090 DATA 2,4,5,7,9,11,13,14,9,1
1,13,14,16,17,19
2100 DATA 21,23,24,23,21,19,17,1
6,17,19,21,19,17,16,14,12
2110 DATA 11,19,16,12,14,19,16,1
2
2120 DATA 26,23,19,21,23,21,19,2
1,19,21,19,18,18
2130 DATA 26,-1,2,7,11,26,23,19,
16,0,4,7,12,16,19,16
2140 DATA 24,-3,0,6,9,24,21,18,1
4,-1,2,6,11,14,18,14
2150 DATA 23,-5,-1,4,7,23,19,16,
12,-3,0,4,9,12,16,12
2160 DATA 21,-6,-3,2,6,21,18,14,
11,-5,-1,2,7,19,14,11
2170 DATA 9,11,12,15,16
2180 DATA 19,14,19,23,26,23,19,2
3,24,21,18,21
2190 DATA 19,14,19,22,26,22,19,2
2,24,21,18,21
2200 DATA 7,-17,-15,-14,-12,-10,
-8,-6,-5,19,22,21,19,17,16,14
2210 DATA 13,-15,-13,-11,-10,-8,
-6,-4,-3,25,28,26,25,22,21,19
2220 DATA -7,2,4,5,7,9,11,13,14,
2,5,4,2,0,-1,-3
2230 DATA -4,11,12,14,16,18,20,2
1,23,-4,-1,-3,-4,-7,-8,-10
2240 DATA -12,21,28,26,24,23,21,
19,17,2,9,7,5,4,2,0
2250 DATA -1,19,26,24,23,21,19,1
7,16,0,7,5,4,2,0,-1
2260 DATA -3,17,24,23,21,19,17,1
6,14,-1,5,4,2,0,-1,-3
2270 DATA -4,16,23,21,20,17,16,1
4,12,-3,0,-1,-3,-5,-7,-8
2280 DATA -10,10,14,12,10,9,7,5,
4,5,7,9,10,12,14,16
2290 DATA 14,13,14,13,14
2300 DATA 14,2,4,5,7,9,11,13,14,
9,11,13,14,16,17,19
2310 DATA 21,23,24,26,28,26,24,2
3,21,19,17,16,14,12
2320 CLS : STOP
3000 PRINT BRIGHT 1;"
                      ESTA A OUV
IR A VERSAO SPECTRUM
                           DA SO
NATA PARA PIANO
                              K5
45 DE MOZART
"
3010 PRINT AT 16,0; BRIGHT 1;"
       TEMPO DE EXECUCAO
   2 MINUTOS E 58 SEGUNDOS
"
3020 GO TO 500
9980 SAVE "MOZART" LINE 1
```

SIMPLIFIQUE OS COMANDOS DO SEU ZX MICRODRIVE

Autor: António Gomes Nunes
FUNCHAL

Todos os possuidores do ZX Microdrive são obrigados a executar um verdadeiro exercício de digitação sempre que pretendem carregar ou gravar um programa. Por exemplo o comando LOAD tem o seguinte formato, assumindo que se está a carregar a partir do drive n.º 1:

LOAD *«m»;1;«nome do programa»

Uma vez carregado o programa que a seguir se apresenta, o comando LOAD passa a fazer-se simplesmente com:

* L «nome do programa»

o que representa uma substancial redução do número de teclas a premir.
Este programa, que utiliza a potencialidade da ROM do Interface 1 que nos permite definir novos comandos, compõe-se de duas partes, uma em BASIC e outra em linguagem máquina.
Digite a parte em BASIC, apresentada seguidamente, e grave-a com SAVE *«m»;1;«run»LINE 1.

```
1 CLEAR 65373: LOAD *«m»;1;«novcom»CODE : CLOSE II 0:
POKE 23735, 94: POKE 23736,255
2 PRINT «Novos comandos:»''«*L LOAD»'«*S SAVE»'«*M
MERGE»'«*V VERIFY»'«*O CAT»'«*R NEW & RUN»
```

Quanto à parte em linguagem máquina, utilize o seguinte programa auxiliar para a sua entrada e gravação:

```
  1 REM Programa auxiliar para entrada do código máquina
 10 CLEAR 65373: LET c=0: RESTORE 100
 20 FOR i=65374 TO 65535
 30 READ a: LET c=c+a: POKE i,a
 40 NEXT i
 50 IF c<>19482 THEN PRINT «Erro num DATA — corri-
    gir»: STOP
 60 SAVE *«m»;1;«novcom»CODE 65374,162
 70 VERIFY *«m»;1;«novcom»CODE 65374,162
 80 PRINT «Código máquina OK»
100 DATA 254,92,194,240,1,215,32,0
110 DATA 246,32,254,115,40,22,254,108
120 DATA 40,28,254,118,40,34,254,109
130 DATA 40,36,254,99,40,38,254,114
140 DATA 40,62,24,222,253,203,124,238
150 DATA 205,199,255,195,54,8,253,203
160 DATA 124,230,205,199,255,195,165,8
170 DATA 253,203,124,254,24,244,253,203
180 DATA 124,246,24,238,33,214,92,54
190 DATA 1,35,54,0,35,54,2,215
200 DATA 32,0,254,13,40,7,254,58
210 DATA 40,3,205,30,6,195,169,4
220 DATA 215,32,0,205,183,5,195,149
230 DATA 10,62,77,50,217,92,215,32
240 DATA 0,215,140,28,215,24,0,223
250 DATA 202,35,7,215,241,43,62,10
260 DATA 167,237,66,218,76,6,120,177
270 DATA 202,76,6,62,1,50,214,92
280 DATA 175,50,215,92,237,67,218,92
290 DATA 237,83,220,92,215,24,0,195
300 DATA 35,7
```

Para testar o programa, faça NEW e RUN (lembre-se de que a parte em BASIC foi gravada com o nome «run»).
Devem então aparecer no écran os novos comandos, os quais se referem ao Microdrive n.º 1.
*1 (LOAD), *s (SAVE), *m (MERGE) e *v (VERIFY) devem ser seguidos do nome do programa, admitindo ainda as variantes habituais (CODE,DATA LINE).
E pronto, a partir de agora pode esquecer os complicados formatos destes comandos, que obrigavam a uma considerável Ginástica de dedos!
Por último refira-se que o reconhecimento destes novos comandos mantém-se activo desde que o programa em linguagem Máquina permanece inalterado na memória, e que o valor da variável do sistema VECTOR não seja modificado. Por isso, mesmo depois de carregar outro programa em BASIC com *1 «. . .» os novos comandos mantém-se activos enquanto não fizer NEW. MAs mesmo depois de fazer NEW pode reactivar os comandos com POKE 23735,94:POKE 23736,255 desde que a parte em linguagem máquina não tenha sido alterada.

NO CLUBE Z 80

(Os mais vendidos)

1 — DEUS EX MACHINA
2 — MACHINE CODE TUTOR
3 — SHERLOCK HOLMES
4 — TURMOIL
5 — TRAVEL WITH TRASHMAN
6 — THE INVERNO
7 — BEACH HEAD
8 — KNIGHT LORE
9 — DARTES
10 — CAVELON E VOLCANO

TOP 10 EM INGLATERRA

1 — GHOST BUSTERS
2 — AIR WOLF
3 — MATCH DAY
4 — KNIGHT LORE
5 — 3D STAR STRIKE
6 — ELITE
7 — STAFF OF KARNATH
8 — SELECT
9 — HUNCHBACK
10 — DOOMDARK REVENGE

GRAVAÇÃO DE PROGRAMAS NO SEPCTRUM, SEM HEADER

Autor: Hugo Assumpção

«Todos sabem que um programa, quando gravado pelo Spectrum, tem duas partes: o cabeçalho ou header e o corpo.

O cabeçalho, como já foi visto nas revistas anteriores n.º 18, pág. 14 e n.º 21, pág. 16, contém indicações para o computador, sobre o tipo de programa, nome, comprimento, onde começa a correr, etc.

É necessário que o computador saiba exactamente onde, como e o que colocar na memória.

Por isso usa o header.

Já é do nosso conhecimento haver programas sem header. Como? Substituindo o cabeçalho. Como? Instruindo o computador, por meio de programas em código máquina, do tipo, comprimento e início do programa a carregar.

Por isso é necessário saber correctamente o início e comprimento. De outra forma é muito difícil conseguir um programa sem cabeçalho.

Aqui vão duas rotinas em ASSEMBLER para SAVE e LOAD sem header.

SAVE:

1 — Carregar o acumulador com um byte marcador eg. 255	ld a, 255
2 — Carregar o registro IX com o endereço de início do programa	ld IX, xxxx
3 — Carregar o registro de com o n.º de bytes a gravar	ld de, xxxx
4 — Chamar a rotina da ROM no endereço 1218	call 1218
5 — Regressar ao basic	ret

LOAD:

1 — Activar carry flag	scf
2 — Carregar o acumulador com o byte marcador eg. 255	ld a, 255
3 — Carregar IX com o endereço de início	ld IX, xxxx
4 — Carregar de com o n.º de bytes a chamar	ld de, xxxx
5 — Chamar a rotina da ROM no endereço 1366	call 1366
6 . Retorno ao basic	ret

SAVE		LOAD	
ld a, xx (255)	62,xx	scf	55
ld IX, xxxx	221,33,xx,xx	ld a,xx (255)	62,xx
ld de, xxxx	17,xx,xx	ld IX,xxxx	221,33,xx,xx
call 1218	205,194,4	ld de,xxxx	17,xx,xx
ret	201	call 1366	205,86,5
		ret	201

Repare que as duas rotinas têm partes iguais.

APLICAÇÃO:

Ex.: Para Save ou Load de um Screen sem header, teríamos o seguinte programa BASIC, para guardar o C.M.:

```
10 DATA 55,62,255,221,33,0,64,17,0,27,205,0,0,201
20 FOR N=23300 TO 23313 : READ A : POKE N,A : NEXT N
   RUN
```

Colocada a rotina no Printer Buffer podemos usá-la fazendo:

```
30 REM SAVE
40 POKE 23311,194 : POKE 23312,4 : RANDOMIZE USR
   23301 : STOP
50 REM LOAD
60 POKE 23311,86 : POKE 23312,5 : RANDOMIZE USR
   23300 : STOP
   SAVE = GO TO 30     LOAD = GO TO 50
```

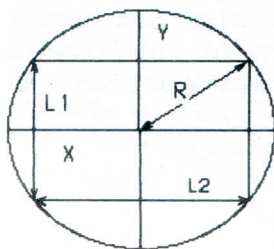

RESPOSTA AO DESAFIO

```
# EQUACOES DO SEGUNDO GRAU #

        2
12 X  -56 X + 34 = 0

        56+√1504
X1  =  ----------
           24

        56-√1504
X2  =  ----------
           24

# EQUACOES DO SEGUNDO GRAU #

        2
25 X  + 67 X = 0

        -67+√4489
X1  =  -----------
            50

        -67-√4489
X2  =  -----------
            50

# EQUACOES DO SEGUNDO GRAU #

        2
-56 X  + 45 = 0

        0-√10080
X1  =  ----------
           112
```

```
               0+√10080
       %2  =  ──────────
                 112

  20 CLS
  30 PRINT BRIGHT 1;"  # EQUACOE
S DO SEGUNDO GRAU #  "
  40 INPUT "COEFICIENTE DE %↑2 (
a)   ";a
  50 IF a=0 THEN GO TO 40
  60 PRINT '''TAB (ABS SGN a+LEN
 STR$ a+4);"2"
  70 IF a=1 THEN PRINT TAB 4;" %
 ";: GO TO 100
  80 IF a=-1 THEN PRINT TAB 4;"-
 % ";: GO TO 100
  90 PRINT TAB 3;a;" % ";
 100 INPUT "COEFICIENTE DE %↑1 (
b)   ";b
 110 IF b=0 THEN GO TO 160
 120 IF b>0 THEN PRINT "+ ";
 130 IF b=1 THEN PRINT "% ";: GO
 TO 160
 140 IF b=-1 THEN PRINT "- % ";:
 GO TO 160
 150 PRINT b;" % ";
 160 INPUT "COEFICIENTE DE %↑0 (
c)   ";c
 170 IF c=0 THEN PRINT "= 0": GO
 TO 210
 180 IF c>0 THEN PRINT "+ ";
 190 IF c<0 THEN PRINT "- ";ABS
c;" = 0": GO TO 210
 200 PRINT c;" = 0"
 210 LET d=b*b-4*a*c
1000 REM PRIMEIRA RAIZ
1010 PRINT AT 11,5;"%1 =";AT 10,
11;-b*SGN a;
1020 IF a<0 THEN PRINT "-√";d: G
O TO 1040
1030 PRINT "+√";d
1040 LET y=0
1050 IF a<0 AND b<0 THEN LET y=1
1060 IF a<0 AND b>0 THEN LET y=-
1
1070 FOR q=1 TO LEN STR$ d
1080 LET h=y+SGN b+LEN STR$ b+q
1090 PRINT AT 9,12+h;"_"
1100 NEXT q
1110 FOR n=1 TO 4+h
1120 PRINT AT 11,9+n;"—"
1130 NEXT n
1140 PRINT AT 12,12+INT ((h/2)-L
EN STR$ ABS (2*a)/2);ABS (2*a)
2000 REM SEGUNDA RAIZ
2010 PRINT AT 16,5;"%2 =";AT 15,
11;-b*SGN a;
2020 IF a<0 THEN PRINT ;"+√";d:
GO TO 2040
2030 PRINT "-√";d
2040 FOR q=1 TO LEN STR$ d
2050 LET h=y+SGN b+LEN STR$ b+q
2060 PRINT AT 14,12+h;"_"
2070 NEXT q
2080 FOR n=1 TO 4+h
2090 PRINT AT 16,9+n;"—"
2100 NEXT n
2110 PRINT AT 17,12+INT ((h/2)-L
EN STR$ ABS (2*a)/2);ABS (2*a)
3000 REM OPCAO CONTIN./TERMINAR
3010 PRINT #0; BRIGHT 1;"    PRI
MA <s> PARA TERMINAR       "
3020 PAUSE 0
3030 IF INKEY$<>"s" AND INKEY$<>
"S" THEN GO TO 20
3040 STOP
9000 REM udg
9010 RESTORE
9020 FOR a=1 TO 3: READ a$
9030 FOR t=0 TO 7: READ b
9040 POKE USR a$+t,b
9050 NEXT t: NEXT a
9060 DATA "p",0,0,0,255,0,0,0,0
9070 DATA "i",0,68,170,16,16,170
,68,0
9080 DATA "r",1,1,66,34,36,20,24
,8
9090 GO TO 20
9990 SAVE "R.Eq.2gr." LINE 9000
```

DESAFIO — Resposta ao n.º 26, pág. 18

António M. Bastos Pereira

```
10 INPUT "RAIO=";R,"LADO=";L
20 IF L>2*R OR L+R<0 THEN PRINT AT 10,0;"DA-
   DOS INCORRECTOS ...ENTER" : PAUSE 0 : RUN
30 LET AREA=L*2*SQR (R*R-L*L/4)
40 PRINT AT 10,0;"AREA=     ";AREA;"  U.M.↑2"
50 PAUSE 0 : RUN
```

Demonstração:
Teorema de Pitágoras

$$x^2 + y^2 = R^2 (1)$$

uma vez que y = L 1/2 em que L1 é o lado do rectângulo e R é o raio de circunferência.
Substituindo em (1) teremos:

$$x^2 + \frac{L1^2}{4} = R^2 \ (===) \ x = \pm\sqrt{R^2 - \frac{L1^2}{4}} \text{ tendo } x = L2/2$$

considerando a Área do Rectângulo = L1 x L2

obtemos a Área = $L1 * L2 \times \sqrt{R^2 - L1^2/4}$

DESAFIO PROPOSTO:

«UM PAR DE NÚMEROS NATURAIS DIZ-SE AMIGÁVEL SE A SOMA DOS DIVISORES DE UM DELES (EXCEPTO ELE PRÓPRIO) É IGUAL AO OUTRO E VICE-VERSA (por exemplo 220 e 284).
Estabelecer um programa que determine mais pares de números amigáveis!

NOVOS PROGRAMAS

?'' ''

GILLIGANS GOLDS

O objectivo deste jogo é apanhar o ouro existente em cada quadro e depositá-lo no carrinho de mão existente no primeiro quadro.É preciso evitar os inimigos, ou seja, os outros bonecos, pois eles se nos alcançarem matam-nos. Evitar também as vagonetes de mina: um acidente com elas é mortal. Para nos defendermos dos ladrões, teremos que pegar a picareta e caminharmos contra eles, ou então pegar no ouro e deixá-lo cair em cima deles. Para evitar as vagonetes, temos que procurar no túnel em que eles rodam, sítios com uma seta.
Aí, basta carregar na tecla para subir e na de apanhar objectos teremos evitado a vagonete. Nos quadros em que o ouro esteja atrás de paredes, teremos que pegar na pá e começar a escavar a parede. Ao passar de um quadro para o outro, e para evitar andar de trás para a frente com o ouro, podemos levar o carrinho de mão connosco, bastando para isso carregar na tecla para apanhar objectos e numa de direcção.

TECLAS: Q — Esquerda P — Subir
W — Direita ENTER — Descer
SPACE — Apanhar, Lançar

BLACK HAWK

O objectivo deste jogo consiste em destruir as bases do inimigo, assim como os seu radares, aviões, helicópteros, mísseis, etc.... Nós tripulamos um dos mais avançados aviões até agora construídos. Mas o inimigo é em grande número, por isso, este jogo é um jogo que essencialmente desafia a rapidez do jogador assim como os seus reflexos. Os quadros são alternados ora com o avião, ora com uma mina. Para disparar (com a mina), basta carregar no botão de disparo e, ao mesmo tempo, num dos botões de direcção. Seguidamente largam o botão de disparo e quando quisermos fixar a mina, largamos o botão de direcção.

TECLAS: O — Esquerda 0 — Disparo
P — Direita

* O programa fornece bastantes instruções sobre os objectivos a destruir, armamento, etc....

VU-FILE 2

Programa de arquivo idêntico ao primeiro VU-FILE, mas com uma vantagem de suportar 64 caracteres em cada linha. As instruções e operações definidas no VU-FILE 2 são as mesmas que no VU-FILE 1.

BACK PACKERS

O objectivo deste jogo é percorrer o labirinto à procura de seres aliegenes e recolhê-los. Devemos evitar as bolas que nos tiram energia sempre que chocam connosco. Precisamos recolher também as bolas com setas, que nos poderão indicar direcção a seguir, assim como as chaves, que servem para abrir portas e barreiras ou barras de dinamite. Para afastar as bolas que nos aparecem em cada túnel podemos disparar uma arma laser, que as destrói.

TECLAS: Q — Subir O — Esquerda
A — Descer P — Direita
M — Disparo/Apanhar objectos

GRAND PRIX MANAGER

Este jogo consiste na simulação de um grande campeonato automobilístico de velocidade. Você vai tomar o lugar à frente de uma equipa e terá que a gerir e comandar, tendo para isso que contratar mecânicos, pilotos, consertar automóveis e, o mais importante, ganhar dinheiro para as diversas etapas do campeonato.
O programa fornece as instruções necessárias para que se possa efectuar todo o tipo de provas, desde o contrato de pilotos até ao tipo de carros a escolher.

LAZY JONES

Este programa é novo no género, e um dos melhores actualmente no mercado do ZX SPECTRUM. Consiste basicamente de um écran no qual existem 18 portas, todas elas conduzindo a uma sala de vídeo ou a dependências utilitárias (quartos, w.c., etc.). Para que o Jones, que nós comandamos, possa entrar em cada um dos quartos e desenvolver a actividade inerente a cada um deles, tem que se carregar na tecla «M». Ao começar o jogo o Jones aparece no piso do meio, e para evitar os guardas de cada um dos pisos, ele pode, ou entrar no elevador (tecla «M») e deslocar-se para o piso inferior ou superior, ou saltar por cima dos guardas, usando a tecla «Q». Não se pode entrar mais do que uma vez num mesmo quarto.
No jogo podem-se escolher o número de vidas que se quiser, entre 1 e 9.

TECLAS: Q — Cima O — Esquerda M — Disparo/Entrar
A — Baixo P — Direita

CYCLONE

Este jogo pertence à mesma série do já famoso «TLL». Desta vez a aventura é com um helicóptero, que tem que andar a recolher cestos (grades) e salvar os sobreviventes do ciclone (bónus extra), devendo regressar o helicóptero à base depois de ter cumprido a missão. Há que ter cuidado com o vento, visto que com muito o helicóptero fica descontrolado e com o risco de se despenhar. O vento aumenta quando nos aproximamos do ciclone. O jogo também contém um mapa, onde podemos ver as posições do helicóptero assim como as do ciclone. O programa fornece instruções acerca da aparelhagem de controlo de bordo, assim como as teclas a utilizar.

TECLAS: 1 - Subir O - Esquerda X - Mudar de direcção
Q - Descer P - Direita M - Mapa
N - Mudança de vista

APPLE JAM

Trata-se de um jogo de acção, constituído por um único écran onde se move um homenzinho cuja missão é apanhar com a boca as gotas de compota e as maçãs que caem de cima. Sempre que uma gota ou uma maçã cair no fundo do écran, um rato come-a e aumenta de tamanho. Ao fim de comer 3 itens, o rato sobe ao piso superior, onde está o homem, e tenta deitá-lo abaixo matando-o. A única defesa dele é entrar no elevador e fazer com que ele caia em cima do rato. Para o fazer, basta calcular o tempo que o elevador leva a subir e descer e carregar na tecla da esquerda, «5», que o elevador faz o resto. Sempre que o rato morre debaixo do elevador, o écran volta ao princípio, repetindo-se o mesmo processo, num nível superior e com mais intervenientes: abelhas, moscas, etc.... Sempre que o homem estiver muito gordo, ele terá que ir à sauna, bastando para isso carregar na tecla da direita, «8».

TECLAS 5 - Esquerda H - Hold (para o jogo)
8 - Direita

BEAR GEORGE

Trata-se de um dos melhor jogos para o SPECTRUM, tanto em questão de gráficos, como no jogo em si. O objectivo do jogo é conduzir o urso George para debaixo das macieiras de modo que ele coma o máximo de maçãs possível. Tem que se evitar os esquilos, que nos atiram com bolotas, que nos provocam uma grande «dor de cabeça» e também perda de peso. No fundo do écran existe um gráfico que nos indica o peso. Depois de se passar o primeiro quadro, temos de conduzir o urso George de um lado para o outro do écran, devendo evitar os esquiadores, que nos podem fazer perder vidas. O terceiro quadro é constituído por uma caverna que o urso George tem que atravessar, evitando as aranhas. Para as evitar temos que passar por baixo delas rapidamente (tecla para acelerar).
Por fim o urso George chega à caverna onde irá hibernar. Se as maçãs que comeu chegarem para o alimentar durante a hibernação, o urso George voltará ao primeiro quadro, repetindo-se o mesmo esquema de jogo, com ligeiras modificações e num nível superior. Se não chegarem as maçãs que comeu, o urso George morre, terminando deste modo o jogo.

TECLAS: Q — Subir a cabeça (d) I — Esquerda
A — Subir a cabeça (e) P — Direita
M — Acelerar o passo

CIRCUS

Trata-se de uma aventura muitíssimo boa, que consegue coordenar o jogo com estratégia, a acção com a aventura e o raciocínio com o dinamismo. Como prefácio, podemos dizer que esta aventura começou com um aviso para o dono do circo de que ele não deveria dar mais nenhum espectáculo, senão... É claro que o dono ficou assustado, e mais ainda quando começam a acontecer coisas estranhas no circo: mortes, roubos, etc....
A nosso missão é descobrir o culpado e acabar com esta onda de fenómenos inexplicáveis.
Na aventura deverá utilizar termos o mais parecidos possível com um verdadeiro diálogo, tais como:

Up — U North — N West — W
Down — D South — S East — E etc....

SYSTEM 15 000

Este programa é uma simulação de «MODEM», em que o SPECTRUM terá que comunicar com outros computadores via telefónica, bastando para isso marcar um determinado número e esperar que ele estabeleça comunicação. Depois de estabelecida teremos que receber mensagens, descobrir o seu verdadeiro sentido, responder e continuar o mais possível a chamada, estabelecendo deste modo uma melhor comunicação entre o SPECTRUM e o outro computador.

OLYMPIMANIA

Este é um novo programa da série do «PYMANIA», as loucas aventuras do «PI». Mais concretamente este jogo trata dos Jogos Olympicos malucos com este famoso e engraçado personagem, que é o «PI». Com efeito, este simpático personagem vai ter que realizar um determinado número de provas olímpicas com vista a uma qualificação (medalha de ouro, prata ou bronze). As provas podem ser: salto de obstáculos, esqui, hipismo, natação, etc.... Mas, para dificultar um pouco as provas, existem obstáculos pelo caminho (e não são poucos...), assim como um inimigo em cada quadro. Para evitar os obstáculos e o inimigo, o «PI» terá que saltar, acelerar ou parar, consoante o caso.
BOA SORTE!
Para mover o «PI», poder-se-á utilizar o «Joystick» ou as seguintes teclas:

6 — Parar/Travar 8 — Saltar 0 — Avançar/Acelerar

PSYTRAXX

Este jogo é dos mais divertidos e difíceis que existem no mercado. Trata-se de mais um da série do «MICROBOT», mas cujo objectivo é bastante mais diverso: 1 microandroide tem que apanhar cartões-chave, passar para chegar a uma CPU muito especial: trata-se da CPU de um robot que domina um império — o império DROID — e que se tornou um autêntico tirano, construindo robots para o ajudarem a montar o império sobre o seu domínio, e que lhe obedeçam cegamente. Para chegar a CPU, teremos portanto que evitar esses guardas, que poderão ter as mais diversas formas: microships, condensadores, transistores, resistores, etc.... e teremos também que lutar com a confusão gerada pelo imperador e provocada pelo grande número de divisões do palácio do imperador — cerca de 100 quadros diferentes!

A — Subir O — Esquerda M — Disparar
Z — Descer P — Direita H — Stop action
Q-T — Apanhar, largar objectos

DRAGONSBANE

Mais um jogo de arcádia da QS. O objectivo deste novo jogo é destronar um feiticeiro que reina num temeroso castelo, protegido por terríveis seres e feitiços, os quais obedecem cegamente ao feiticeiro. Para nos protegermos destes perigos, teremos várias armas descritas no jogo, assim como objectos que formos apanhando no caminho.
O programa descreve as teclas que são utilizadas no jogo, assim como a sua posição.

MERCADO Z80

O MERCADO Z80 É UMA SECÇÃO DO CLUBE Z80 QUE EMPRESA AOS SEUS SÓCIOS PROGRAMAS E LIVROS (SPECTRUM) PARA CONSULTA E MELHOR CONHECIMENTO/APROVEITAMENO DE MICROCOMPUTADORES.

— PROGRAMAS: Todos os jogos e utilitários que existem no CLUBE Z80, excepto programas de cópia e programas com direitos de autor (Ex.: "Cálculo de Estruturas").

— LIVROS: Cerca de 40 títulos diferentes.

COMO TORNAR-SE SÓCIO DO MERCADO Z80?

Para poder ter em sua posse 5 cassetes ou livros durante um mês, basta enviar-nos um depósito de Esc. 2000$00 (garantia de que os materiais nos serão devolvidos em estado de conservação e funcionamento idêntico àquele em que foram enviados).
Ao mesmo tempo, deverá remeter-nos a quantia de Esc. 1000$00 que será a base da sua "Conta-Corrente". Essa quantia servirá para pagar as suas despesas:

— Taxa de utilização dos produtos: 250$00 (referente a 5 unidades, entre livros e programas).

— Instruções dos programas (no caso de o sócio não as devolver, debitar-lhe-emos 5$00 por folha).

— Embalagem Postal: 20$00 a 30$00 (no caso de o pedido ser feito via CTT.

— Portes dos CTT's: 40$00 a 80$00 (no caso de o pedido ser feito via CTT.

IMPORTANTE!

— O depósito de 2000$00 pertence integralmente ao sócio desde que os materiais por ele utilizados nos sejam devolvidos nas mesmas condições em que saíram do CLUBE Z80. Assim, quando o sócio desistir do MERCADO Z80, essa quantia ser-lhe-á entregue.
Em caso de extravio, danos ou avarias dos materiais, o sócio pagará o valor comercial dos respectivos produtos (a descontar no depósito de 2000$00).

— No caso de os produtos seguirem via CTT, o sócio não pagará para levantar a encomenda. As despesas serão pagas por nós, no momento da expedição, e debitadas ao sócio (a descontar no depósito de 1000$00).

— Quando as suas despesas estiverem a atingir os 1000$00 avisá-lo-emos, e o sócio deverá renovar essa quantia de modo a cobrir despesas seguintes.

— A taxa de utilização dos produtos é fixa — 250$00. Ela refere-se ao conjunto de 5 unidades. (Pagará sempre 250$00 mesmo que peça só uma unidade).

QUE PRODUTO E QUE QUANTIDADES?

O sócio nunca pode pedir mais do que 5 unidades de cada vez (entre livros e cassetes). Quanto a livros não poderemos empresar mais do que um. Assim, o sócio poderá pedir:

— 5 cassettes
ou
— 4 cassettes + 1 livro

QUAL O TEMPO DE UTILIZAÇÃO?

O sócio poderá ficar com os produtos durante 1 MÊS, no máximo. Findo esse período, deverá devolvê-los ao CLUBE Z80.
O MERCADO Z80 só atenderá dois pedidos por mês, para cada sócio.

COMO FAZER O PEDIDO?

Numa carta, escreva pelo menos 10 títulos (por ordem de prioridade). Se os 5 primeiros não estiverem disponíveis, enviaremos os outros evitando grandes esperas de produtos que estejam em circulação.
Ao devolver os produtos, inclua uma carta com o pedido seguinte.

IMPORTANTE! O SÓCIO SÓ PODERÁ EFECTUAR UM NOVO PEDIDO JUNTAMENTE COM A DEVOLUÇÃO DO MATERIAL CORRESPONDENTE AO PEDIDO ANTERIOR (ou depois, se preferir).

Se estiver interessado no MERCADO Z80, faça já o seu 1.º pedido, enviando 3000$00 e o cupão abaixo devidamente preenchido.

INSCRIÇÃO NO MERCADO Z80

NOME ______________________________

ENDEREÇO ______________________________

CÓDIGO POSTAL ______________________________

TELEFONE ______________

ENVIO 3 000$00 (2 000$00 como garantia de que devolverei os produtos em boas condições + 1 000$00 para a minha «Conta Corrente» em:

☐ Cheque n.º ______________ ☐ Vale Postal ☐ Dinheiro

Banco ______________ N.º ______________ ______________

Data ___/___/___ *Assinatura do Sócio* ______________________________

É SÓCIO DO CLUBE Z80? ☐ SIM ☐ NÃO

CLUBE Z80

INSCRIÇÃO COMO ASSOCIADO

O **CLUBE Z80** está aberto a todos os utilizadores de microcomputadores.

A intenção de associar os entusiastas das micro-máquinas, é exclusivamente a de permitir:

1 — PUBLICAÇÃO DE UM JORNAL MENSAL, onde sejam publicados programas de uso geral ou específico como no caso da educação.

2 — PROMOVER TROCAS DE PROGRAMAS, e trocas de experiências; tanto no caso do Software (programação), como no caso do Hardware (electrónica).

3 — PROMOVER DESCONTOS NA AQUISIÇÃO DE PROGRAMAS.

4 — LANÇAR CURSOS DE PROGRAMAÇÃO EM BASIC — PASCAL OU OUTRAS LINGUAGENS E DIVULGAR O USO DE LINGUAGEM MÁQUINA.

NOME ....

IDADE .... COMPUTADOR TIPO ....

PROFISSÃO ....

ENDEREÇO ....

.... TELEF. ....

ASSINATURA ANUAL — Esc. 1 500$00 ☐

ASSINATURA SEMESTRAL — Esc. 750$00 ☐

CHEQUE OU VALE DO CORREIO

N.º ....

BANCO ....

DATA ..../......../........

JÁ SÓCIO ☐

NOVO SÓCIO ☐ → A partir do mês de .... (inclusive)